

# SISTEM KOMUNIKASI PADA *AUTONOMOUS SURFACE VEHICLE*

Fadel Muhammad Imbang  
Teknologi Telekomunikasi Fakultas Ilmu  
Terapan  
Bandung, Indonesia  
fadlmhammad@student.telkomuniversity  
.ac.id

Dadan Nur Ramadan, S.Pd.,  
M.T.  
Teknologi Telekomunikasi  
Fakultas Ilmu Terapan  
Bandung, Indonesia  
dadannr@telkomuniversity.ac.  
id

Asep Mulyana, S.T., M.T.  
Teknologi Telekomunikasi  
Fakultas Ilmu Terapan  
Bandung, Indonesia  
asepmulyana@telkomuniversity.ac.id

## Abstrak

Indonesia merupakan negara maritim dengan potensi wisata bahari cukup besar, mendorong inovasi otomasi dalam teknologi sistem navigasi perkapalan. Terdapat berbagai teknologi dan metode untuk sistem otomasi navigasi ini antara efisiensi yang lebih baik antara lain menggunakan metode sistem pengenalan lingkungan di permukaan laut atau ASV

Dalam Proyek Akhir ini dirancang dan direalisasikan suatu model RC Boat yang dilengkapi dengan sistem navigasi otomatis menggunakan teknologi HSV yaitu menggunakan metode pendeteksian warna (melalui kamera) menggunakan balon berwarna hijau dan merah di permukaan air untuk kendali arah (belok kiri/kanan) serta deteksi rintangan. Data hasil deteksi ini ditransfer dari Raspberry Pi (pada perahu) ke laptop (di daratan) melalui Wi-Fi dengan frekuensi 2.4 GHz. Sebagai pembandingan terhadap sistem kontrol navigasi otomatis dilakukan pula sistem navigasi manual (RC Boat) yang dilakukan dari daratan untuk kendali arah dan kecepatan.

Hasil perancangan RC Boat menunjukkan performa baik dalam mode *autonomous* dan manual. Meskipun menghadapi *delay* pada *image processing*, analisis Wireshark memberikan wawasan untuk optimasi jaringan. Hasil uji akurasi untuk sistem otomatis maupun manual menunjukkan kesesuaian 100%. Sementara rata-rata *delay* untuk sistem manual sistem otomatis yaitu 28.86429043 ms.

**Kata kunci :** ASV, HSV, OpenCV, Wi-Fi, Delay, Image Processing.

## Abstract

Indonesia is a maritime country with considerable marine tourism potential, encouraging automation innovation in shipping navigation system technology. There are various technologies and methods for this navigation automation system between better efficiency, among others, using the method of environmental recognition systems on the sea surface or ASV.

In this Final Project, a RC Boat model equipped with an automatic navigation system using HSV technology is designed and realized, namely using a color detection method (through a camera) using green and red balloons on the water surface for direction control (left/right turn) and obstacle detection. The detection data is transferred from the Raspberry Pi (on the boat) to the laptop (on land) via Wi-Fi with a frequency of 2.4 GHz. As a comparison to the automatic navigation control system, a manual navigation system (RC Boat) is also carried out from land for direction and speed control.

The RC Boat design results show good performance in both *autonomous* and manual modes. Despite the delay in image processing, Wireshark analysis provides insights for network optimization. Accuracy test results for both automatic and manual systems show 100% compliance. While the average delay for the automatic system manual system is 28.86429043 ms.

**Keywords :** ASV, HSV, OpenCV, Wi-Fi, Delay, Image Processing.

## 1. PENDAHULUAN

Indonesia adalah negara maritim dengan dua pertiga wilayahnya terdiri dari lautan dan garis pantai sepanjang 81.290 km, terpanjang kedua di dunia setelah Kanada. Ini memberikan potensi besar dalam wisata bahari untuk meningkatkan ekonomi, kesejahteraan, lingkungan, dan budaya. Untuk mendukung pariwisata bahari, diperlukan kapal atau bangunan apung dengan teknologi canggih. Inovasi teknologi perkapalan dan digital kemaritiman sangat penting agar mampu bersaing dengan negara maju,

dengan memasukkan teknologi digital otonom (otomatisasi) untuk efisiensi [1].

RC Boat adalah sarana transportasi air penting yang berkembang menjadi kendaraan tanpa awak untuk meningkatkan efisiensi dan navigasi. *Autonomous Surface Vehicle* (ASV) adalah contoh RC Boat tanpa awak yang dapat bergerak otomatis, membutuhkan sistem komunikasi handal untuk pertukaran data antar sensor, kendali dan navigasi, bagian dari *Internet of Things* (IoT) [2].

Proyek ini merancang RC *Boat* dengan navigasi otomatis menggunakan pemodelan warna HSV dan OpenCV untuk pengolahan citra *realtime*. Komunikasi menggunakan *Wi-Fi* 2.4 GHz. Kunci perancangan adalah kemampuan RC *Boat* menghindari rintangan dan bergerak aman serta efisien..

## 2. DASAR TEORI

### 2.1 *Autonomous Surface Vehicle (ASV)*

*Autonomous Surface Vehicle (ASV)* adalah perahu RC *Boat* tak berawak yang dapat berpindah dari satu titik ke titik lain menggunakan navigasi *waypoint*. ASV telah digunakan dalam operasi militer sejak Perang Dunia II, dan kini negara maju berlomba mengembangkannya [3]. ASV memungkinkan pemantauan perairan berbahaya secara *realtime*, menjangkau area yang sulit dijangkau manusia. Meskipun manfaatnya besar dalam observasi perairan, perkembangan ASV di Indonesia masih tertinggal dan memerlukan pengembangan lebih lanjut.

### 2.2 *Python*

*Python* adalah bahasa pemrograman populer yang digunakan dalam aplikasi *web*, pengembangan perangkat lunak, ilmu data, dan *machine learning*. Developer menggunakan *Python* karena efisien, mudah dipelajari, gratis, dan kompatibel dengan berbagai *platform*, sehingga mempercepat pengembangan [4].

### 2.3 *Arduino IDE*

*Arduino IDE* adalah *software* untuk membuat dan mengunggah *sketch* pemrograman ke *board* *Arduino*. Dibuat dengan bahasa pemrograman *JAVA* dan dilengkapi library *C/C++ (wiring)*, *Arduino IDE* mempermudah operasi *input/output* [5]. Dengan antarmuka yang sederhana dan intuitif, pengguna dari berbagai tingkat keahlian dapat membuat, mengembangkan, dan menguji proyek elektronik dengan mudah.

### 2.4 *Hue Saturation and Value (HSV)*

*HSV (Hue, Saturation, dan Value)* adalah model warna yang menyerupai warna pandangan manusia. Dalam model ini, warna dijelaskan sebagai campuran *non-linear* dari *RGB (Red, Green, Blue)* [6]. *Hue* mengindikasikan posisi warna dari merah ke hijau, *saturation* menggambarkan intensitas warna dari merah hingga merah muda, sementara *value* mengacu pada tingkat kecerahan dari warna hitam hingga putih.

### 2.5 *OpenCV*

*Open Computer Vision (OpenCV)* adalah pustaka *open source* untuk pengolahan citra, memungkinkan komputer memiliki kemampuan visual mirip manusia. Dikembangkan oleh Intel, *OpenCV* menyederhanakan pemrograman terkait citra dengan berbagai algoritma visi komputer dan modul deteksi objek. *OpenCV*

dapat digunakan dengan bahasa pemrograman seperti *C, C++, Java, dan Python* [7].

### 2.6 *Thonny*

*Thonny IDE* adalah lingkungan pengembangan terintegrasi yang digunakan untuk bahasa pemrograman *Python*. Dirancang khusus untuk pendidikan dan pembelajaran, *Thonny* menyediakan antarmuka yang sederhana, sehingga sangat cocok bagi pemula yang baru memulai belajar pemrograman [8].

### 2.7 *Wireshark*

*Wireshark* merupakan sebuah paket analisis yang dapat mengambil paket pada jaringan dan mempresentasikan paket data secara detail. *Wireshark* dapat menganalisis paket jaringan sebagai alat ukur yang digunakan untuk memeriksa apa yang terjadi dalam jaringan [9].

### 2.8 *Parameter QoS*

*Quality of Service (QoS)* adalah metode untuk mengukur dan mendefinisikan karakteristik jaringan. *QoS* memastikan data yang dikirim melalui jaringan mencapai tujuan dengan kualitas memadai, menggunakan parameter seperti *latency (delay), jitter, packet loss, dan throughput* [10].

### 2.9 *Radio Control*

*Radio Control* merujuk pada gelombang radio untuk mengendalikan perangkat atau kendaraan dari jarak jauh. Teknologi ini terdiri dari pemancar (*transmitter*) yang dipegang operator dan penerima (*receiver*) pada perangkat. Pada proyek ini, digunakan komunikasi *Wi-Fi* 2.4 GHz dengan standar 802.11 b/g/n, yang umum untuk komunikasi nirkabel.

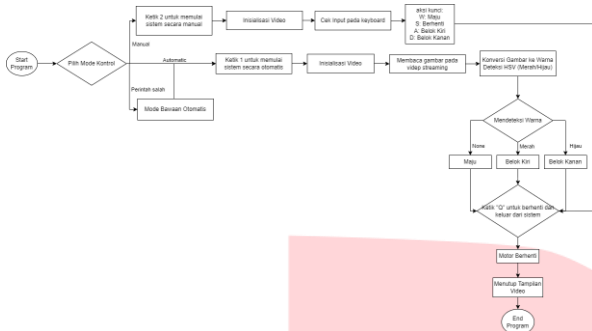
## 3. PERENCANAAN SISTEM KOMUNIKASI

### 3.1 *Deskripsi Proyek Akhir*

Proyek ini merancang sistem komunikasi untuk *Autonomous Surface Vehicle (ASV) Radio Control Boat* dengan navigasi otomatis di atas air. Sistem menggunakan model warna HSV dan library *OpenCV* untuk deteksi warna dan pengolahan citra *realtime*. Komunikasi *Wi-Fi* 2.4 GHz dengan standar 802.11 b/g/n menghubungkan Laptop (*main processing*), *Raspberry Pi (server)*, dan *Arduino UNO (motor Thruster)* untuk mengatur perahu.

Analisis *QoS* dilakukan dengan *Wireshark* untuk memastikan komunikasi yang optimal, menangkap dan menganalisis paket data guna memastikan sistem responsif dalam navigasi otomatis. *Wireshark* menyediakan fitur seperti inspeksi paket mendalam dan filter tampilan untuk memantau lalu lintas jaringan *Wi-Fi ASV RC Boat*.

### 3.2 Alur Perancangan Sistem



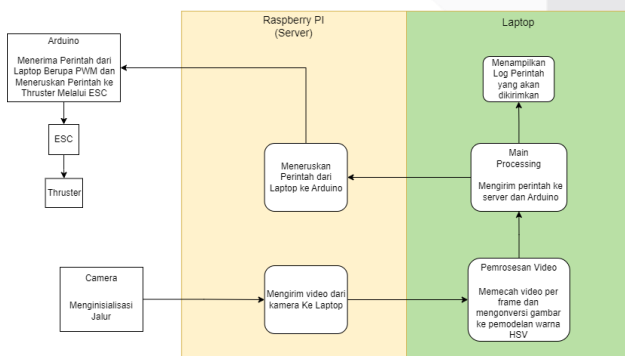
Gambar 3. 1 Alur Sistem RC Boat

Gambar 3.1 menjelaskan alur kerja sistem *Autonomous Surface Vehicle (ASV)*. Saat sistem dijalankan, pengguna memilih mode kontrol: angka 1 untuk otomatis atau angka 2 untuk manual. Perintah yang tidak valid akan mengalihkan sistem ke mode otomatis secara *default*.

Di mode otomatis, kamera memulai video, memecah gambar per *frame*, dan mengirimnya ke laptop melalui server untuk diproses. Laptop mengonversi gambar ke format warna HSV dan OpenCV, kemudian menyatukannya kembali menjadi video MP4. Jika warna hijau terdeteksi, laptop mengirim perintah ke Arduino untuk belok kanan; jika merah, belok kiri; jika tidak ada, perintah untuk maju. Untuk keluar, pengguna mengetik "Q" yang menghentikan motor dan menutup tampilan video.

Di mode manual, kamera juga memulai video, dan pengguna mengontrol perahu dengan *keyboard*: "W" untuk maju, "S" untuk berhenti, "A" untuk kiri, dan "D" untuk kanan. Perintah "Q" akan menghentikan sistem dan menutup tampilan video, sama seperti di mode otomatis.

### 3.3 Perancangan Sistem Komunikasi



Gambar 3. 2 Blok Diagram Perancangan Sistem

Gambar 3.2 menjelaskan sistem *Autonomous Surface Vehicle (ASV) Radio Control Boat* yang

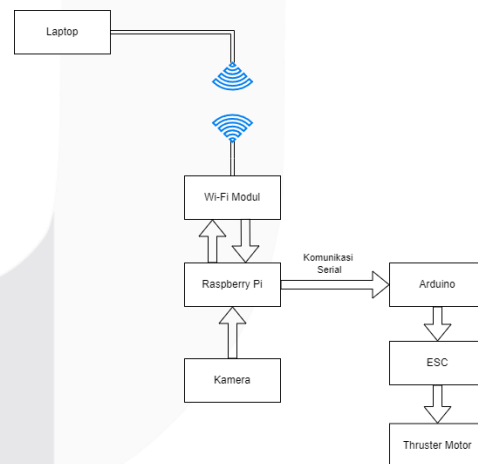
terdiri dari tiga komponen utama: Laptop, *Raspberry Pi*, dan Arduino, yang terhubung secara *realtime*.

Laptop berfungsi sebagai unit pemrosesan utama, mengendalikan perahu, memproses video untuk navigasi berdasarkan balon merah dan hijau, dan menampilkan tampilan kamera. Proses konversi warna dilakukan dengan *thresholding* untuk mengidentifikasi warna merah (HSV *lower*: (0, 120, 70), *upper*: (10, 255, 255)) dan hijau (HSV *lower*: (40, 40, 40), *upper*: (80, 255, 255)). Gambar yang telah di *threshold* diproses di *Raspberry Pi*.

*Raspberry Pi* bertindak sebagai server yang meneruskan data dari kamera dan Arduino ke laptop melalui *Wi-Fi*. Balon merah dan hijau dipilih untuk jalur navigasi karena visibilitas tinggi dan kontras jelas.

Arduino mengontrol baling-baling perahu (*Thruster*) melalui ESC berdasarkan perintah dari *Raspberry Pi* dan mengirimkan status perintah serta tegangan. Sistem dimulai dengan kamera yang merekam video jalur dan mengirim data ke *Raspberry Pi*. *Raspberry Pi* meneruskan data video ke laptop, yang memprosesnya menggunakan model warna HSV dan OpenCV. Jika warna merah terdeteksi, laptop mengirim perintah PWM ke Arduino melalui *Raspberry Pi*. Arduino kemudian mengirim perintah ke *Thruster* melalui ESC, memastikan perahu bergerak sesuai instruksi.

### 3.4 Perancangan Komunikasi Boat



Gambar 3. 3 Blok Diagram Komunikasi Boat

Komunikasi pada *RC Boat* menggunakan *Wi-Fi* 2.4 GHz dengan standar 802.11 b/g/n, yang dikembangkan oleh IEEE pada 2009 dengan kecepatan maksimum 100 Mbps. Standar ini memungkinkan perangkat nirkabel untuk berkomunikasi secara cepat dan stabil tanpa kabel, memberikan fleksibilitas dan mobilitas yang lebih besar.

Untuk memastikan kinerja jaringan optimal, analisis *delay* menggunakan Wireshark diperlukan. Wireshark menganalisis paket data untuk mengukur *delay* antara pengiriman perintah dan penerimaan

respons, membantu mengidentifikasi *bottleneck* dan melakukan optimasi. Analisis ini memastikan kontrol ASV yang responsif dan efisien dalam aplikasi *realtime*.

## 4. ANALISIS SIMULASI SISTEM

### 4.1 Deskripsi Simulasi Sistem

Bab ini menganalisis hasil simulasi sistem dan komunikasi yang telah dirancang. Simulasi menggunakan perangkat lunak seperti VNC untuk akses nirkabel, Arduino IDE untuk Arduino UNO, VS Code untuk pemrosesan citra, dan Thonny untuk server.

Simulasi meliputi tahap di darat dan air, dengan pengujian sistem *autonomous* dan manual. Di darat, diuji integrasi komponen dan komunikasi antara laptop, Raspberry Pi, dan Arduino. Di perairan, perahu diuji dalam kondisi nyata untuk menilai kinerja sistem.

Keberhasilan sistem *autonomous* ditentukan oleh kemampuan perahu mendeteksi dan mengikuti jalur warna hijau dan merah dengan akurasi tinggi, menggunakan kamera dan *library* OpenCV. Keberhasilan sistem manual ditentukan oleh kemampuan perahu merespons perintah navigasi dengan cepat dan tepat melalui konfigurasi di Arduino IDE dan VS Code, tanpa *delay* signifikan.

### 4.2 Pengujian Sistem Komunikasi ASV RC Boat

Pengujian sistem perahu dilakukan di kolam renang GKU Telkom *University* pada tanggal 6 Juni 2024. Lokasi ini menjadi lingkungan ideal untuk mengevaluasi performa sistem navigasi perahu dalam kondisi yang terkendali. Fokus utama dari pengujian ini adalah untuk menguji kemampuan sistem dalam dua mode operasi utama: *autonomous* dan manual.



Gambar 4. 1 Pengujian Perahu di Kolam Renang GKU

Pengujian di kolam renang dipilih karena:

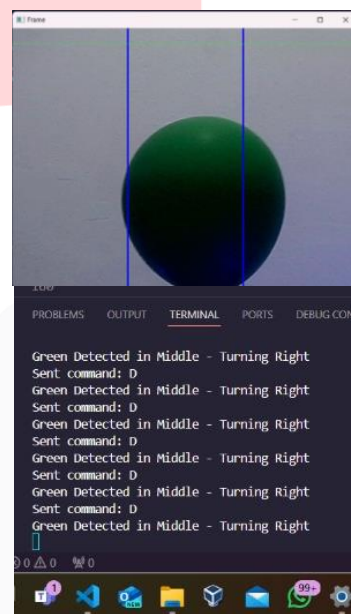
1. Lingkungan Terkendali: Meminimalkan variabel eksternal untuk data yang konsisten.
2. Keamanan dan Kemudahan: Mengurangi risiko kerusakan dan memudahkan pengamatan serta penyesuaian.
3. Pengulangan Uji coba: Memungkinkan validasi dan pengembangan sistem lebih mendalam.

Data pengujian meliputi akurasi navigasi, kecepatan, stabilitas, dan respons terhadap rintangan. Pengujian di GKU Telkom *University* penting untuk mengembangkan teknologi navigasi otomatis yang andal, dengan potensi aplikasi di berbagai lingkungan perairan seperti danau, sungai, dan laut. Lingkungan terkendali memastikan teknologi memenuhi standar kinerja dan keselamatan sebelum diuji di kondisi yang lebih kompleks.

### 4.3 Hasil Pengujian Sistem

Hasil parameter sistem yang akan ditampilkan terbagi kedalam 2 perbandingan, yaitu perbandingan hasil pengujian dengan mode otomatis dan hasil pengujian dengan mode manual.

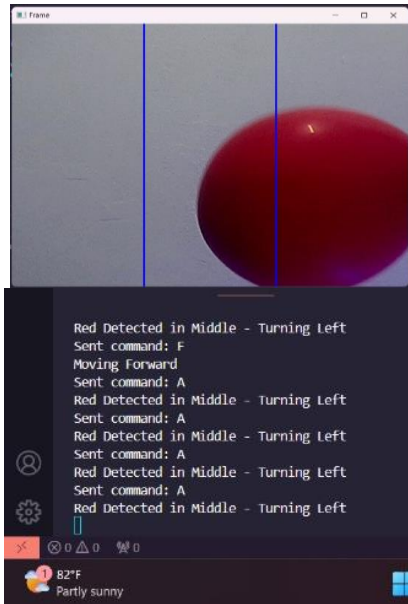
#### 4.3.1 Hasil Pengujian Sistem Autonomous



Gambar 4.3.1 Kamera Mendeteksi Warna Hijau

Pada Gambar 4.3.1, kamera berhasil mendeteksi balon hijau pada pengujian pertama. Sistem *autonomous* perahu menunjukkan tingkat keberhasilan tinggi dalam mendeteksi jalur dan berfungsi baik meskipun dalam kondisi kurang ideal. Tampilan di VS Code memperlihatkan deteksi balon hijau oleh kamera dan instruksi untuk belok kanan.

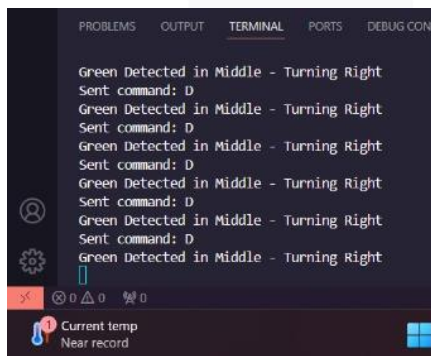
Pengujian menunjukkan perahu mengenali penanda dengan akurasi baik di berbagai kondisi pencahayaan. Respons perahu terhadap instruksi navigasi juga cepat dan tepat, memastikan navigasi efisien dan penghindaran rintangan yang baik. Secara keseluruhan, sistem *autonomous* ini dapat diandalkan untuk navigasi otomatis.



Gambar 4.3.2 Kamera Mendeteksi Warna Merah

Pada Gambar 4.3.2, kamera berhasil mendeteksi balon merah di jalur navigasi perahu. Sistem visi komputer mengidentifikasi warna merah sebagai sinyal untuk berbelok kiri, menerjemahkannya menjadi instruksi navigasi yang akurat. Perahu secara otomatis berbelok kiri saat balon merah terdeteksi, dan proses ini dapat dipantau secara *realtime* di VS Code.

### 4.3.2 Hasil Pengujian Sistem Manual



Gambar 4.3. 3 Pengujian Sistem secara Manual

Gambar 4.3.3 menunjukkan hasil pengujian sistem navigasi kapal secara manual. Pada gambar terdapat instruktur untuk mengarahkan RC Boat berbelok ke kanan dengan menekan huruf "D" pada keyboard melalui antarmuka di VS Code. Sistem merespons perintah manual ini dengan tepat, menggerakkan kapal sesuai instruksi. Pengujian ini penting untuk memastikan keandalan sistem dalam mode manual dan mengidentifikasi potensi masalah atau keterlambatan respons. Gambar ini tidak hanya menunjukkan akurasi sistem dalam mengikuti perintah manual tetapi juga berfungsi sebagai bagian dari proses validasi dan kalibrasi untuk memastikan operasi yang aman dan efisien.

## 4.4 Hasil Analisis QoS

Analisis QoS melibatkan penilaian *delay*, *throughput*, *jitter* dan *packet loss* menggunakan *software* Wireshark. Data yang digunakan untuk analisis ini dapat ditemukan pada Gambar 4.4.1.

Measurement	Captured	Displayed	Marked
Packets	67568	67568 (100.0%)	—
Time span, s	194.129	194.129	—
Average pps	348.1	348.1	—
Average packet size, B	1262	1262	—
Bytes	85299556	85299556 (100.0%)	0
Average bytes/s	439 k	439 k	—
Average bits/s	3515 k	3515 k	—

Gambar 4.4 1 Wireshark Capture File Properties

### 4.4.1 Hasil Analisis QoS

Analisis *delay* merupakan waktu yang diperlukan untuk paket data untuk berpindah dari sumber ke tujuan. Berdasarkan penjelasan diatas didapatkan rumus untuk menghitung rata-rata delay yaitu:

$$\text{Rata - rata delay} = \frac{\sum \text{Delay tiap paket}}{n} \quad (1)$$

Dimana:

- $\sum \text{Delay}$  tiap paket = total dari semua *delay* yang diukur untuk setiap paket yang dikirim.
- $n$  = jumlah paket yang dikirim dan di ukur *delay*.

Hasil dari analisis *delay* dalam satuan s (*second*). Untuk mengonversi ke dalam ms (*millisecond*) dikali 1000. Berdasarkan Gambar 4.4.1 mendapatkan rata-rata *delay* 28.86429043 ms.

### 4.4.2 Hasil Analisis Throughput

Analisis *throughput* merupakan Jumlah data yang berhasil dikirimkan melalui jaringan dalam periode waktu tertentu. *Throughput* yang tinggi menunjukkan efisiensi jaringan yang baik. Rumus menghitung *throughput* sebagai berikut:

$$\text{Throughput} = \frac{\text{Jumlah Bytes}}{\text{Time Span}} \quad (2)$$

Hasil dari perhitungan *throughput* harus dikali 8 untuk mengubah satuan dari KiloByte ke Kilobit. Hasil analisis *throughput* menggunakan Wireshark berdasarkan data pada gambar 4.4.1 dapatkan rata-rata *throughput* sebesar 3515 Kb atau 439 KB .

### 4.4.3 Hasil Analisis Jitter

Analisis *jitter* merupakan variasi dalam waktu pengiriman paket data. *Jitter* yang tinggi dapat menyebabkan masalah pada aplikasi yang sensitif terhadap waktu, seperti video *streaming* dan VoIP. Rumus menghitung *jitter* sebagai berikut:

$$\text{Jitter} = |(T3 - T2) - (T2 - T1)| \quad (3)$$

Dimana :

- T = Waktu kedatangan paket

Hasil analisis *Jitter* menggunakan Wireshark berdasarkan data pada gambar 4.4.1 dapatkan total

*jitter* yaitu 283.314 ms dan rata-rata *jitter* sebesar 0.161340547 s atau 161.3405467 ms.

#### 4.4.4 Hasil Analisis *Packet Loss*

Analisis *packet loss* merupakan persentase paket data yang hilang selama transmisi. *Packet loss* yang rendah sangat penting untuk menjaga integritas data dan kualitas layanan. Rumus menghitung *packet loss* sebagai berikut:

$$[(\text{paket dikirim} - \text{paket diterima}) : \text{paket dikirim}] \times 100 \quad (4)$$

Hasil analisis *packet loss* menggunakan Wireshark berdasarkan data pada gambar 4.4.1 dapatkan total *packet loss* yaitu 100%. Ini menandakan tidak adanya paket data yang hilang selama proses pengiriman.

## 5. KESIMPULAN

Berdasarkan hasil perancangan, pengujian, dan analisis yang dilakukan, dapat disimpulkan bahwa sistem RC Boat menunjukkan performa yang baik baik dalam mode otomatis maupun manual. Dalam mode otomatis, perahu berhasil mengikuti jalur yang ditentukan dengan akurasi yang memadai, sementara dalam mode manual, operator dapat mengontrol perahu dengan efektif. Hal ini membuktikan bahwa desain dan implementasi sistem telah memenuhi tujuan fungsionalnya.

Namun, pengujian juga mengungkapkan adanya masalah *delay* dalam pemrosesan gambar pada mode otomatis. *Delay* ini disebabkan oleh keterbatasan sumber daya komputasi saat menggunakan metode pemrosesan gambar kompleks seperti deteksi warna dengan OpenCV. Masalah ini mempengaruhi responsivitas sistem, terutama dalam situasi yang memerlukan kecepatan tinggi. Di sisi lain, mode manual berfungsi dengan lancar dan mengalami sedikit *delay*.

Analisis *delay* menggunakan Wireshark memberikan wawasan berharga mengenai kinerja jaringan dan komunikasi sistem. Pengukuran *delay* ini tidak hanya mengidentifikasi waktu transmisi data antar komponen, tetapi juga memungkinkan evaluasi faktor-faktor yang mempengaruhi kinerja keseluruhan. Dengan analisis yang akurat, dilakukan perbaikan pada infrastruktur jaringan, optimasi protokol komunikasi, dan penyesuaian perangkat keras, sehingga meningkatkan efisiensi dan responsivitas sistem. Penggunaan Wireshark membantu memastikan bahwa sistem RC Boat dapat beroperasi dengan optimal dalam berbagai kondisi operasional.

## DAFTAR PUSTAKA

- [1] S. M. P. D. E. G. L. P. S. M. D. E. I. P. S. A. S. M. P. D. E. H. Y. S. M. S. S. M. H. S. M. I. H. N. D.-I. P. Andi Haris Muhammad, PEDOMAN KONTES KAPAL CEPAT TAK BERAWAK NASIONAL (KKCTBN), Jakarta: Balai Pengembangan Talenta Indonesia, Pusat Prestasi Nasional, Sekretariat Jenderal Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, 2023.
- [2] D. D. A. R. Andi Annisa Reski Febrina, "Sistem Komunikasi Autonomous Boat Dan Ground Control Station Guna Mendukung Penelitian Autonomous Fish Feeder Swarm Boat Di Laboratorium Inacos Universitas Telkom," *e-Proceeding of Applied Science*, vol. 9, no. 1, p. 297, 2023.
- [3] T. BEATRIZ, Desain Kendali Sistem Geram Bow Thruster Pada Autonomous Surface Vehicle Dengan Menggunakan Metode Sliding Mode Control, Surabaya: Institut Teknologi Sepuluh November, 2020.
- [4] S. Y. Raihan Muhammad, "PENERAPAN PEMROGRAMAN PYTHON DALAM MENENTUKAN WAKTU OVERHOUL KONDENSOR TURBIN UAP," *Jurnal Konsersi Energi dan Manufaktur*, vol. 8, no. 1, pp. 49-57, 2023.
- [5] L. N. Z. H. Jauhari Arifin, "PERANCANGAN MUROTTAL OTOMATIS MENGGUNAKAN MIKROKONTROLLER ARDUINO MEGA 2560," *Jurnal Media Infotama*, vol. 12, no. 1, 2016.
- [6] M. A. R. B. R. Haidar Azmi Rabbni, "Perbandingan Ruang Warna RGB dan HSV dalam Klasifikasi Kematangan Biji Kopi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6, pp. 2243-2248, 2021.
- [7] E. M. Y. Tengku Cut Al-Saidina Zulkhaidi, "Pengenalan Pola Bentuk Wajah dengan OpenCV," *JURTI*, vol. 3, no. 2, pp. 181-185, 2019.
- [8] P. R. P. E. D. N. Muhammad Abdul Hadi, "RANCANG BANGUN MODUL PRAKTIKUM SISTEM EMBEDDED BERBASIS RASPBERRY PI (PENGONTROLAN DASAR LED, LED DOT-MATRIX, DAN SEVEN SEGMENT DISPLAY)," *Jurnal SPEKTRUM*, vol. 8, no. 2,

2021.

- [9] WASKITO, "ANALISA MALWARE PADA TRAFFIC JARINGAN DATA MENGGUNAKAN WIRESHARK," *Telkom UNiversity Open Library*, pp. 5-12, 2019.
- [10] N. R. S. Muhamad Hasbi, "ANALISIS QUALITY OF SERVICE (QOS) JARINGAN INTERNET KANTOR PUSAT KING BUKOPIN DENGAN MENGGUNAKAN WIRESHARK," vol. 12, no. 1, pp. 17-23, 2021.
- [11] I. G. I. M. O. G. G. M. N. V. Romasella Tri Novita, "Analisis Keamanan Wifi Menggunakan Wireshark," *Jurnal Elektro Smart*, vol. 1, no. 1, 2021.