

# Implementasi ROS dan Optimasi Identifikasi Warna *Buoy* Dengan Metode YOLOv5 Pada Miniatur *Autonomous Surface Vehicle*

Anugerah Ekha Gusti Audryadmaja<sup>1</sup>, Joko Endrasmono<sup>2</sup>, Zindhu Maulana Ahmad Putra<sup>3</sup>, Lilik Subiyanto<sup>4</sup>, Muhammad Khoiril Hasin<sup>4</sup>, Mohammad Basuki Rahmad<sup>5</sup>, Isa Rachman<sup>6</sup>, Agus Khumaidi<sup>7</sup>, Yuning<sup>8</sup>, Ryan Yudha Adhitiya<sup>9</sup>, Dimas Pristovani Riananda<sup>10</sup>

e-mail: [anugerah.ekha@student.ppns.ac.id](mailto:anugerah.ekha@student.ppns.ac.id)<sup>1</sup>, [endrasmono@ppns.ac.id](mailto:endrasmono@ppns.ac.id)<sup>2</sup>, [zindhu@ppns.ac.id](mailto:zindhu@ppns.ac.id)<sup>3</sup>

<sup>123</sup> Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya, JL. Teknik Kimia, Keputih, Sukolilo, Surabaya, Indonesia

## Informasi Artikel

### Riwayat Artikel

Diterima 15 Mei 2024

Direvisi 23 Mei 2024

Diterbitkan 31 Mei 2024

### Kata kunci:

*Autonomous surface vehicle*  
Robot Operating System  
YOLOv5

### Keywords:

*Autonomous surface vehicle*  
*Robot Operating System*  
YOLOv5

## ABSTRAK

*Autonomous Surface Vehicles (ASV)* telah menjadi fokus penelitian yang populer karena aplikasinya yang luas. Tantangan utama dalam pengembangan ASV adalah deteksi dan identifikasi objek di permukaan air, seperti *buoy*, dengan cepat dan akurat. Penelitian ini mengintegrasikan *Robot Operating System (ROS)* dengan algoritma YOLOv5 untuk mendeteksi *buoy* berwarna, dengan tujuan mengidentifikasi varian YOLOv5 yang menawarkan kinerja komputasi ringan dan akurasi yang tinggi untuk aplikasi *real-time* pada ASV. Berbagai varian YOLOv5 (YOLOv5s, YOLOv5m, dan YOLOv5L) dievaluasi berdasarkan kinerja deteksi dan penggunaan sumber daya komputasi. Hasil menunjukkan semua model YOLOv5 berfungsi dengan baik dalam ekosistem ROS. YOLOv5m memiliki performa deteksi terbaik dengan akurasi 90%, diikuti oleh YOLOv5s (89,8%) dan YOLOv5L (88,9%). Dalam pengujian *real-time*, YOLOv5L memiliki akurasi terbaik sebesar 87,78%, diikuti oleh YOLOv5m (85%) dan YOLOv5s (80%). Dari segi respons, YOLOv5s adalah yang tercepat dengan waktu respons 36,4ms, diikuti oleh YOLOv5m (77ms) dan YOLOv5L (139ms). Berdasarkan hasil tersebut, YOLOv5m dianggap paling ideal untuk digunakan pada ASV miniatur karena memiliki akurasi yang baik (85%) dan waktu respons yang cepat (77ms). Dengan demikian penelitian ini dapat berkontribusi pada pengembangan teknologi ASV yang lebih efisien dan andal yang dapat digunakan pada misi-misi khusus.

## ABSTRACT

*Autonomous Surface Vehicles (ASVs)* have become a popular research focus due to their wide range of applications. A major challenge in the development of ASVs is the rapid and accurate detection and identification of objects on the water surface, such as buoys. This study integrates the *Robot Operating System (ROS)* with the YOLOv5 algorithm to detect colored buoys, aiming to identify the YOLOv5 variant that offers lightweight computational performance and high accuracy for real-time ASV applications. Various YOLOv5 variants (YOLOv5s, YOLOv5m, and YOLOv5L) were evaluated based on detection performance and computational resource usage. The results showed that all YOLOv5 models performed well within the ROS ecosystem. YOLOv5m exhibited the best detection performance with an accuracy of 90%, followed by YOLOv5s (89.8%) and YOLOv5L (88.9%). In real-time testing, YOLOv5L had the highest accuracy at 87.78%, followed by YOLOv5m (85%) and YOLOv5s (80%). In terms of response time, YOLOv5s was the fastest at 36.4ms, followed by YOLOv5m (77ms) and YOLOv5L (139ms). Based on these findings, YOLOv5m is considered the most ideal for use in miniature ASVs due to its good accuracy (85%) and fast response



*time (77ms). This research contributes to the development of more efficient and reliable ASV technology, which can be used in specialized missions.*

**Penulis Korespondensi:**

Anugerah Ekha Gusti Audryadmaja

Program Studi Teknik Otomasi,

Teknik Kelistrikan Kapal,

Politeknik Perkapalan Negeri Surabaya, Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, Indonesia, Kode Pos 60111

Email: [anugerah.ekha@student.ppns.ac.id](mailto:anugerah.ekha@student.ppns.ac.id)

Nomor HP/WA aktif: +62 8785 9029 171

**1. PENDAHULUAN**

*Autonomous surface vehicle* (ASV) telah menjadi topik penelitian yang menarik dalam beberapa tahun terakhir, terutama karena potensi aplikasi mereka yang luas dalam pemantauan lingkungan, misi penyelamatan, dan operasi kelautan lainnya. ASV adalah kendaraan otonom yang beroperasi di atas permukaan air[1]. Salah satu tantangan utama dalam pengembangan ASV adalah kemampuan untuk mendeteksi dan mengidentifikasi objek di permukaan air, seperti buoy, dengan cepat dan akurat. Deteksi yang tepat sangat penting untuk navigasi yang aman dan efisien, serta untuk pengambilan keputusan otomatis[2].

Penelitian sebelumnya telah menggunakan berbagai metode seperti YOLOv4 dan segmentasi warna untuk deteksi objek pada ASV, namun masih terdapat beberapa kekurangan, terutama dalam hal kecepatan dan akurasi deteksi. Pada YOLOv4 memiliki akurasi yang baik namun memiliki komputasi yang berat sedangkan segmentasi warna memiliki akurasi yang buruk dan perlu di kalibrasi sebelum di gunakan[3]. Beberapa metode yang telah diterapkan tidak cukup efisien untuk aplikasi real-time, khususnya pada ASV berukuran kecil dengan keterbatasan daya komputasi. Inilah gap penelitian yang perlu diatasi, yaitu kurangnya penelitian yang fokus pada penerapan algoritma deteksi objek yang efisien dan akurat dalam lingkungan ROS pada ASV dengan keterbatasan komputasi.

*Robot Operating System* (ROS) adalah kerangka kerja yang banyak digunakan dalam pengembangan sistem robotika karena popularitas dan dukungannya terhadap berbagai sensor dan aktuator [4]. Integrasi ROS dengan algoritma deteksi objek canggih seperti YOLOv5 (*You Only Look Once version 5*) dapat meningkatkan kemampuan deteksi objek pada ASV. YOLOv5 adalah salah satu algoritma deteksi objek tercepat dan paling akurat saat ini, dengan berbagai varian model yang menawarkan *trade-off* antara kecepatan dan akurasi[5].

Penelitian ini berfokus pada implementasi YOLOv5 dalam lingkungan ROS untuk mendeteksi warna *buoy* pada miniatur ASV. Tujuan utama dari penelitian ini adalah untuk mengidentifikasi varian YOLOv5 yang memberikan kinerja komputasi yang ringan serta nilai eror deteksi yang rendah. Dengan demikian, diharapkan dapat ditemukan solusi optimal yang memungkinkan deteksi objek yang efisien tanpa mengorbankan akurasi, yang sangat penting untuk aplikasi *real-time* pada ASV berukuran kecil dengan keterbatasan daya komputasi. Untuk mencapai tujuan ini, peneliti melakukan serangkaian eksperimen yang melibatkan berbagai varian model YOLOv5, yaitu YOLOv5s, YOLOv5m, dan YOLOv5L. Setiap model dievaluasi berdasarkan kinerja deteksi dan penggunaan sumber daya komputasi. Pengujian dilakukan dalam kondisi yang bervariasi untuk memastikan validitas dan reliabilitas hasil yang diperoleh dan dapat diketahui manakah varian yang dapat berjalan dengan baik dan lancar pada ekosistem ROS.

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan teknologi ASV yang lebih efisien dan andal, dengan fokus pada aplikasi deteksi objek yang sangat diperlukan dalam berbagai misi kelautan. Temuan ini juga diharapkan dapat memberikan panduan bagi peneliti dan praktisi dalam memilih model YOLOv5 yang paling sesuai untuk diterapkan pada platform ASV dengan keterbatasan spesifik.



## 2. METODE PENELITIAN

### 2.1 Robot Operating System

ROS (*Robot Operating System*) adalah sebuah kerangka kerja yang bersifat terbuka dan dirancang untuk mendukung pengembangan perangkat lunak robot[6]. ROS menyediakan berbagai perangkat lunak alat (*tools*), pustaka (*libraries*), dan paket (*packages*) yang digunakan untuk mengontrol perangkat keras pada robot[7]. Kemampuan ROS mencakup komunikasi antar mesin atau robot melalui pertukaran pesan, yang memungkinkan koordinasi arah robot dan interaksi dengan perangkat keras tingkat rendah. Berikut logo ROS tersedia pada Gambar 2.



Gambar 2: Logo ROS

Dalam penelitian ini, peneliti memanfaatkan keunggulan ROS untuk mengimplementasikan dan mengoptimalkan deteksi warna *buoy* menggunakan YOLOv5 pada sebuah miniatur *Autonomous surface vehicle* (ASV). Dengan menggunakan ROS, peneliti dapat mengembangkan perangkat lunak dengan mudah, serta menguji berbagai varian model YOLOv5 (seperti YOLOv5s, YOLOv5m, YOLOv5l) untuk menentukan model yang paling efisien dalam hal komputasi dan akurasi deteksi jika berjalan pada lingkungan ROS.

### 2.2 You Only Look Once

Deteksi objek atau biasa disebut dengan *object detection* adalah suatu proses yang digunakan untuk menentukan keberadaan objek tertentu dalam suatu citra digital[8]. YOLOv5 (*You Only Look Once version 5*) adalah algoritma deteksi objek yang sangat populer dan efisien, dirancang untuk mendeteksi dan mengenali objek dalam gambar atau video secara *real-time*. YOLOv5 adalah versi lanjutan dari seri YOLO yang dikembangkan oleh komunitas *Ultralytics*, dengan fokus pada peningkatan kinerja, kemudahan penggunaan, dan fleksibilitas. Algoritma ini terkenal karena kemampuannya untuk memproses gambar dengan cepat sambil mempertahankan tingkat akurasi yang tinggi[9].

YOLOv5 memiliki 3 bagian utama pada arsitekturnya yaitu *backbone*, *neck*, dan *head*. Setiap bagian arsitektur memiliki fungsi dan peranan masing-masing[10]. Pertama *backbone* bagian ini bertanggung jawab untuk mengekstrak fitur dari gambar input. YOLOv5 menggunakan CSPDarknet53 sebagai *backbone*, yang merupakan versi modifikasi dari Darknet53. CSPDarknet53 dirancang untuk meningkatkan efisiensi dengan mengurangi redundansi dalam fitur yang diekstrak. Kedua *Neck* Bagian ini terdiri dari beberapa lapisan konvolusi yang bertujuan untuk menggabungkan fitur pada berbagai skala. YOLOv5 menggunakan *PANet* (*Path Aggregation Network*) sebagai *neck*, yang membantu dalam memperkaya informasi fitur dengan menggabungkan informasi dari lapisan yang berbeda untuk meningkatkan kemampuan deteksi objek kecil maupun besar. Terakhir ialah *Head* bagian ini berperan untuk menghasilkan prediksi *bounding box* dan klasifikasi objek[11]. YOLOv5 menggunakan beberapa lapisan konvolusi untuk menghasilkan tiga set prediksi pada skala yang berbeda, yang memungkinkan deteksi objek pada berbagai ukuran. Berikut adalah gambar arsitektur YOLOv5 yang tersedia pada Gambar 3.

Terdapat beberapa varian yang dimiliki oleh YOLOv5 antara lain adalah YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. Setiap varian YOLOv5 memiliki tingkat komputasi dan kemampuan deteksi yang berbeda-beda seperti YOLOv5s yang memiliki komputasi yang ringan namun memiliki keakuratan deteksi yang lebih rendah dari semua varian sedangkan YOLOv5x memiliki komputasi yang berat namun memiliki kemampuan deteksi yang paling baik dibandingkan dengan varian yang lainnya. Dimana penelitian ini akan berfokus pada respons deteksi pada setiap varian YOLOv5 yang akan dijalankan pada ekosistem ROS.





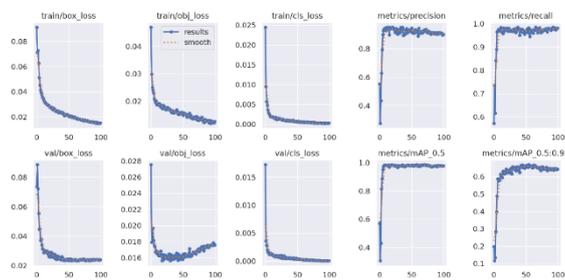
Data set diambil langsung dengan mendorong miniatur ASV yang terpasang web cam Logitech c922 dan laptop didalnya lalu memotret satu persatu objek *buoy* yang dilalui. Dari proses pengambilan data terdapat 1200 foto yang akan digunakan sebagai data set dan telah memenuhi kriteria yang di inginkan. Setelah data set didapatkan maka tahapan selanjutnya ialah proses anotasi yang bertujuan untuk memberi tanda pada setiap objek yang akan di deteksi. Proses *labeling* menghasilkan file yang berisi kelas dan posisi objek, file nantinya akan menjadi acuan dalam proses *training*. Setelah dilakukan anotasi maka data set akan di bagi menjadi 2 bagian yaitu data set digunakan untuk *train* dan validasi dimana data *train* digunakan untuk acuan dalam melatih bobot sedangkan data validasi digunakan untuk memvalidasi hasil dari proses *training*. Selanjutnya ialah proses *training* yang di jalankan dengan bantuan *Google Colaboratoy* dengan mode *runtime* GPU. Sebelum melakukan proses *training* terdapat beberapa parameter yang harus di sesuaikan yaitu *epoch* dan *batch*. Pada penelitian ini seluruh varian YOLOv5 yang akan di uji akan di *training* dengan parameter yang sama dengan *epoch* sebanyak 100 dan *batch* sebesar 16. Selanjutnya dialukan proses *training* dengan masing-masing varian di *training* sebanyak satu kali dengan parameter dan data set yang sama.

Dari hasil *training* yang telah di akukan maka setiap bobot akan diuji secara langsung (*real time*) dengan video yang diambil dari miniatur saat miniatur berjalan meliwati lintasan *buoy* agar diperoleh parameter pengujian yang sama dalam setiap pengujian. Dalam proses pengujian semua bobot akan berjalan di ekosistem ROS yang berjalan pada laptop yang sama dengan laptop yang digunakan pada proses pengambilan data dengan spesifikasi CPU Intel *core i5 gen 10* dengan GPU Nvidia mx330 agar dapat diketahui manakah varian yang sesuai dengan ekosistem ROS dengan spesifikasi tersebut.

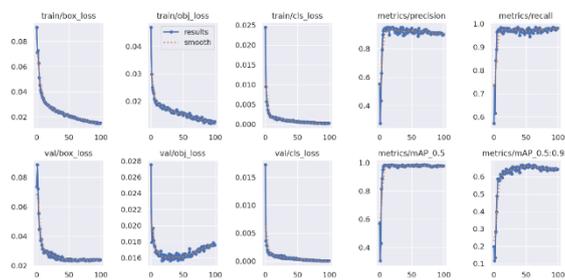
### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil *Training* dan pengujian Bobot

Setelah dilakukan proses *training* maka didapatkan model bobot pada setiap variannya yaitu YOLOv5s, YOLOv5m, dan YOLOv5L. Dari proses proses *training* setiap bobot memiliki nilai *train loss* dan *vall loss* masing-masing. berikut adalah hasil *training* pada masing-masing varian disajikan pada Gambar 5, Gambar 6, dan Gambar 7.

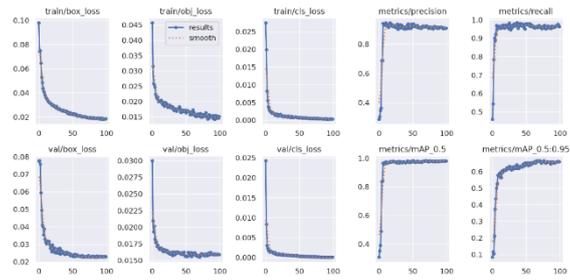


Gambar 5: Hasil *Training* Varian Small



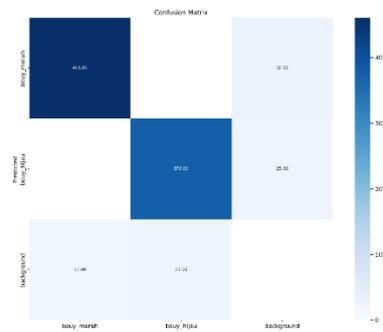
Gambar 6: Hasil *Training* Varian Medium



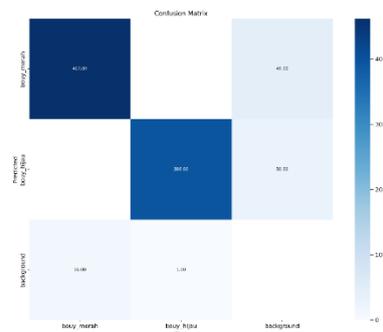


Gambar 7: Hasil *Training* Varian Large

Dari hasil tersebut diketahui bahwa model varian *large* memiliki grafik yang lebih konsisten dan sedikit mengalami lonjakan. Dibanding varian lainya model dengan tipe *large* juga memiliki nilai los yang paling minimal sehingga kemungkinan objek gaga terdeteksi lebih kecil. Namun, perlu diingat bahwa varian ini memiliki kebutuhan komputasi yang tinggi dibanding dengan varian lain. Selanjutnya untuk mengetahui keakuratan bobot dilakukan pengujian menggunakan *konfusi* matriks. Berikut adalah hasil *konfusi* matriks setiap bobot yang di sajian pada Gambar 8, Gambar 9, dan Gambar 10.

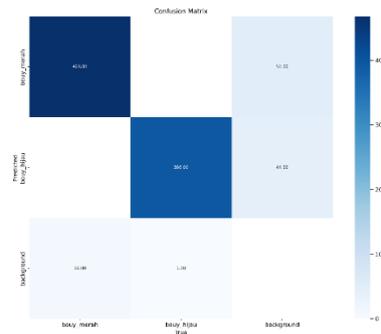


Gambar 8: Konfusi Matrix Varian *Small*



Gambar 9: *Konfusi* Matrix Varian *Medum*





Gambar 10: Konfusi Matrix Varian Large

Dari *konfusi* matriks yang disajikan pada Gambar 8, Gambar 9, dan Gambar 10 dapat dihitung tingkat akurasinya dengan persamaan berikut.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \tag{1}$$

Keterangan;

- *True Positive* (TP):  
Interpretasi: Untuk memprediksi positif dan itu benar.
- *True Negative* (TN):  
Interpretasi: Untuk memprediksi negatif dan itu benar.
- *False Positive* (FP): (Kesalahan Tipe 1)  
Interpretasi: Untuk memprediksi positif dan itu salah.
- *False Negative* (FN): (Kesalahan Tipe 2, kesalahan tipe 2 ini sangat berbahaya)  
Interpretasi: Untuk memprediksi negatif dan itu salah.

Berikut adalah hasil perhitungan *Accuracy* bobot pada setiap varian yolov5 yang diuji pada penelitian ini disajikan pada Tabel I. Dari hasil pengujian menggunakan *konfusi* matriks diperoleh hasil bahwa semua bobot memiliki nilai *accuracy* yang tidak jauh berbeda hanya memiliki selisih kurang dari 2% dengan bobot varian *medium* memiliki Tingkat akurasi paling tinggi sebesar 90.0% dan varian *large* memiliki akurasi paling rendah sebesar 88.9%. walaupun demikian perlu juga di perhatikan bahwa setiap bobot memiliki nilai FN (*False Negatif*), dimana kondisi ini ialah kesalahan paling berbahaya karena objek tidak dapat di prediksi oleh model sehingga seolah - olah objek ini tidak ada, pada model *small* memiliki Tingkat FN yang paling besar sebanyak 40 data dan varian *large* memiliki nilai FN terendah sebesar 11 data saja. Dengan demikian dapat diketahui bahwa setiap model memiliki tingkat akurasi yang tidak jauh berbeda.

TABEL I: ACCURACY BOBOT

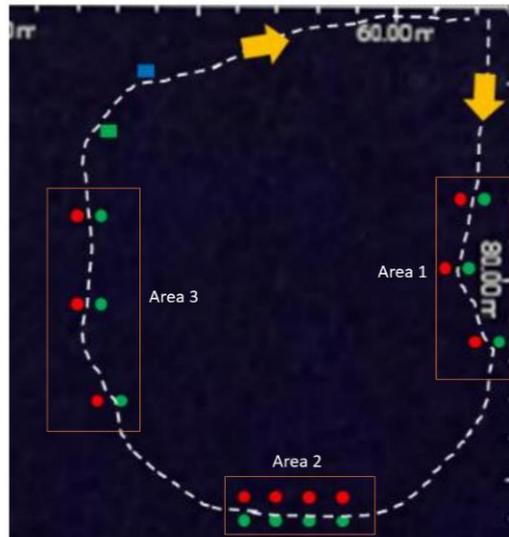
Varian	Accuracy Bobot
Yolov5s	89.8%
Yolov5m	90.0%
Yolov5l	88.9%

### 3.2 Pengujian Bobot Secara *Real Time*

Pengujian *real time* dilakukan dengan cara melakukan uji deteksi secara langsung pada lintasan pengujian ASV guna mengetahui kemampuan setiap model dalam mendeteksi secara langsung dan mengetahui kecepatan respons sistem. Lintasan pengujian terdiri dari susunan *buoy* berwarna merah dan hijau yang membentuk sebuah jalur

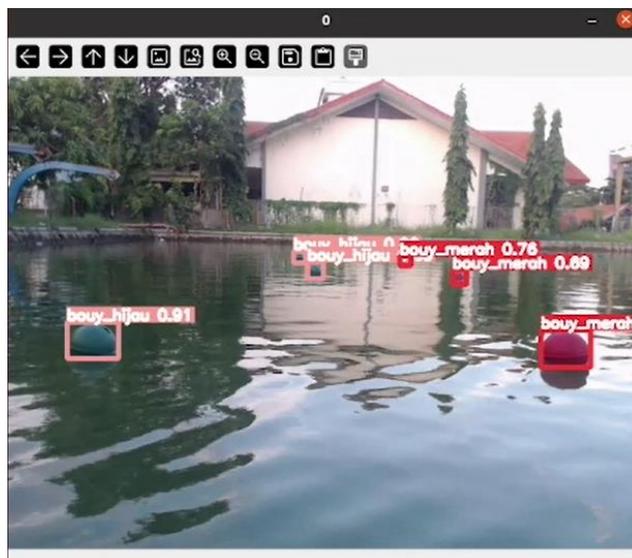


yang terdiri dari 3 area yaitu 2 jalur berkelok dan satu jalur lurus. Untuk memperjelas lintasan pengujian yang di gunakan berikut adalah gambar lintasan pengujian yang di sajian pada Gambar 11.



Gambar 11: Lintasan Pengujian ASV

Pengujian akan dilakukan sebanyak 10 kali pengujian pada masing-masing model YOLOv5 yang dilakukan pada 10 titik pengujian yang telah ditentukan dengan posisi kamera dihadapkan pada lintasan pengujian. Masing-masing model yolo akan di jalan kan pada lingkungan ROS untuk mendeteksi objek, dengan demikian model akan mendeteksi objek *buoy* yang ditangkap oleh kamera seperti yang di sajian pada Gambar 12.



Gambar 12 : Deteksi objek buoy yang ditangkap oleh kamera



TABEL II : PENGUJIAN MODEL YOLOV5S

No.	Respons Time (ms)	Predicted		Real Objek		Akurasi
		Buoy_merah	Buoy_Hijau	Buoy_merah	Buoy_Hijau	
1.	33	3	2	3	3	83.33%
2.	34	3	3	3	3	100.00%
3.	38	4	2	4	5	66.67%
4.	39	3	3	4	4	75.00%
5.	40	3	3	4	4	75.00%
6.	36	2	4	4	4	75.00%
7.	35	2	4	4	4	75.00%
8.	39	3	1	3	3	66.67%
9.	34	3	2	3	3	83.33%
10	36	2	2	2	2	100.00%
<b>Rata - rata</b>	36.4			<b>Rata – rata Error</b>		80.00%

TABEL III : PENGUJIAN MODEL YOLOV5M

No.	Respons Time (ms)	Predicted		Real Objek		Akurasi
		Buoy_merah	Buoy_Hijau	Buoy_merah	Buoy_Hijau	
1.	78	2	3	3	3	83.33%
2.	79	3	3	3	3	100.00%
3.	73	4	2	4	5	66.67%
4.	74	3	3	4	4	75.00%
5.	75	3	3	4	4	75.00%
6.	76`	2	4	4	4	75.00%
7.	78	3	3	4	4	75.00%
8.	77	3	3	3	3	100.00%
9.	79	3	3	3	3	100.00%
10	80	2	2	2	2	100.00%
<b>Rata - rata</b>	77			<b>Rata – rata Error</b>		85.00%

TABEL IV : PENGUJIAN MODEL YOLOV5L

No.	Respons Time (ms)	Predicted		Real Objek		Akurasi
		Buoy_merah	Buoy_Hijau	Buoy_merah	Buoy_Hijau	
1.	137	3	3	3	3	100.00%
2.	137	3	3	3	3	100.00%
3.	136	4		4	5	77.78%
4.	139	2	4	4	4	75.00%
5.	138	3	3	4	4	75.00%
6.	142	2	4	4	4	75.00%
7.	139	2	4	4	4	75.00%
8.	139	3	3	3	3	100.00%
9.	140	3	3	3	3	100.00%
10	152	2	2	2	2	100.00%
<b>Rata - rata</b>	139			<b>Rata – rata Error</b>		87.78%



Untuk lebih detail hasil pengujian secara *real time* data hasil pengujian di sajikan pada tabel II untuk varian Yolov5s, Tabel III untuk varian Yolov5m, dan tabel IV untuk yoloV5. Dari pengujian *real time* dapat diketahui bagaimana performa sebenarnya yang dimiliki oleh masing-masing model. Sesuai yang di sajikan pada tabel II, III, dan IV diketahui varian L memiliki akurasi yang paling tinggi sebesar 87.78%, sedangkan varian M sebesar 85% dan varian S memiliki akurasi yang paling rendah dengan 80%. Untuk respons sistem tercepat di miliki oleh varian s dengan rata-rata respons sebesar 36.4ms, lalu untuk varian m sebesar 77ms dan untuk varian l memiliki respons paling lambat sebesar 139ms. Dengan demikian bahwa setiap varian dapat berjalan di ekosistem ROS dan mendeteksi objek dengan baik.

#### 4. KESIMPULAN

Penelitian ini menunjukkan bahwa penggunaan YOLOv5 memiliki peningkatan performa dibandingkan dengan YOLOv4 yang digunakan pada penelitian sebelumnya, dengan seluruh model YOLOv5 dapat berjalan dengan baik dalam ekosistem ROS. Dari pengujian menggunakan confusion matrix, varian YOLOv5m memiliki performa terbaik dengan akurasi 90%, diikuti oleh YOLOv5s dengan akurasi 89,8%, dan YOLOv5l dengan akurasi 88,9%. Dalam pengujian secara real-time, varian YOLOv5l memiliki akurasi terbaik sebesar 87,78%, diikuti oleh YOLOv5m dengan akurasi 85%, dan YOLOv5s dengan akurasi 80%. Varian yang memiliki respons paling cepat adalah YOLOv5s dengan waktu respons 36,4 ms, diikuti oleh YOLOv5m dengan waktu respons 77 ms, dan varian yang paling lambat adalah YOLOv5l dengan waktu respons 139 ms. Dengan demikian, varian YOLOv5m adalah yang paling ideal untuk digunakan pada miniatur *Autonomous Surface Vehicle* (ASV) karena memiliki akurasi yang cukup baik sebesar 85% dan waktu respons yang cepat di bawah 100 ms, yaitu 77 ms.

#### DAFTAR PUSTAKA

- [1] M. J. Hong and M. R. Arshad, "Modeling and motion control of a riverine autonomous surface vehicle (ASV) with differential thrust," *J. Teknol.*, vol. 74, no. 9, pp. 137–143, 2015, doi: 10.11113/jt.v74.4817.
- [2] F. Romadloni, J. Endrasmono, Z. M. A. Putra, A. Khumaidi, I. Rachman, and R. Y. Adhitya, "Identifikasi Warna Buoy Menggunakan Metode You Only Look Once Pada Unmanned Surface Vehicle," *J. Tek. Elektro dan Komput. TRIAC*, vol. 10, no. 1, pp. 23–29, 2023, doi: 10.21107/triac.v10i1.19650.
- [3] M. B. SHOLAHUDDIN, "Analisa Perubahan Warna Hsv Pada Pengolahan Citra Terhadap Intensitas Cahaya Sebagai Dasar Penerapan Masukan Kontrol Automatic Stacking Crane," pp. 20–25, 2017, [Online]. Available: <https://repository.its.ac.id/45896/>
- [4] A. Jalil, "Robot Operating System (Ros) Dan Gazebo Sebagai Media Pembelajaran Robot Interaktif," *Ilk. J. Ilm.*, vol. 10, no. 3, pp. 284–289, 2018, doi: 10.33096/ilkom.v10i3.365.284–289.
- [5] P. Somaldo, F. A. Ferdiansyah, G. Jati, and W. Jatmiko, "Developing Smart COVID-19 Social Distancing Surveillance Drone using YOLO Implemented in Robot Operating System simulation environment," *IEEE Reg. 10 Humanit. Technol. Conf. R10-HTC*, vol. 2020-Decem, 2020, doi: 10.1109/R10-HTC49770.2020.9357040.
- [6] I. Y. Arulampalam Kunaraj, P. Chelvanathan, Ahmad AA Bakar, "PENGEMBANGAN ROBOT MOBIL MENGGUNAKAN ROS Wea," *J. Eng. Res.*, vol. 1, no. 1, pp. 1–14, 2023, [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK558907/>
- [7] D. Dwiyanto, N. Fath, R. Sepak, and B. Beroda, "Analisa Sistem Komunikasi Data Pada," vol. 3, no. 2, pp. 419–430, 2020.
- [8] W. Yang, Di. Bo, and L. S. Tong, "TS-YOLO: An efficient YOLO Network for Multi-scale Object Detection," *IEEE 6th Inf. Technol. Mechatronics Eng. Conf. ITOEC 2022*, vol. 6, pp. 656–660, 2022, doi: 10.1109/ITOEC53115.2022.9734458.
- [9] A. V. EGA and W. ARDIATNA, "Study on Image Processing Method and Data Augmentation for Chest X-Ray Nodule Detection with YOLOv5 Algorithm," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 2, p. 424, 2023, doi: 10.26760/elkomika.v11i2.424.
- [10] R. Dwiyanto, D. W. Widodo, and P. Kasih, "Implementasi Metode You Only Look Once ( YOLOv5 ) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung," *Semin. Nas. Inov. Teknol.*, vol. 1, no. 1, pp. 102–104, 2022.
- [11] A. Khumaidi, R. Y. Adhitya, D. Wardani, M. R. Fahmi, S. Utomo, and M. D. Khairansyah, "Design of a Fire Spot Identification System in PT . PAL Indonesia Work Area Using," 2023.

