Identifikasi Warna *Buoy* Menggunakan Metode *You Only Look Once* Pada *Unmanned Surface Vehicle*

Faiz Romadloni[1], Joko Endrasmono[2], Zindhu Maulana Ahmad Putra[3], Agus Khumaidi[4], Isa Rachman[5], Ryan Yudha Adhitya[6]
Program Studi Teknik Otomasi, Jurusan Tenkik Kelistrikan Kapal
Politeknik Perkapalan Negeri Surabaya
Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111
E-mail: [1]faizromadloni@student.ppns.ac.id, [2]endrasmono@ppns.ac.id, [3]zindhu@ppns.ac.id, [4]aguskhumaidi@ppns.ac.id, [5]isarachman@ppns.ac,id, [6]ryanyudhaadhitya@ppns.ac.id

***Abstrak*— *Unmanned Surface Vehicle* merupakan kapal permukaan tanpa awak yang dapat beroperasi secara otomatis maupun manual dengan kontrol dari manusia. *Unmannned Surface Vehicle* dilengkapi oleh berbagai sistem seperti sistem komunikasi, sistem propulsi, dan sistem deteksi yang memungkinkannya untuk dapat berlayar dan bernavigasi dengan baik. Salah satu sarana navigasi yang penting dalam dunia pelayaran adalah *buoy* (pelampung suar). *Buoy* memiliki kode warna tertentu yang digunakan sebagai tanda peringatan, larangan, atau perintah bagi kapal yang memasuki area tersebut. Oleh karena itu, identifikasi warna *buoy* secara cepat, tepat, dan *real-time* sangat dibutuhkan untuk mengurangi potensi kecelakaan di wilayah laut, terutama pada *Unmanned Surface Vehicle* yang tidak memiliki awak kapal. Pada penelitian ini digunakan metode *You Only Look Once* untuk mengidentifikasi warna *buoy*. Metode *You Only Look Once* dipilih karena dapat mendeteksi objek secara *real-time* dengan kecepatan yang tinggi. Dari hasil penelitian didapatkan nilai *Mean Average Precision* sebesar 99,3% dan nilai *average loss* sebesar 0,2383. Algoritma ini juga telah diuji pada intensitas cahaya yang berbeda beda. dimana semua pengujian menghasilkan rata rata nilai deteksi sebesar 98,8% untuk *buoy* merah dan 100% untuk *buoy* hijau. Sehingga dapat disimpulkan bahwa metode ini memiliki nilai yang baik dalam deteksi maupun akurasi.**

Kata Kunci*— *Unmanned Surface Vehicle, Buoy, You Only Look Once, Warna, Real-Time

I. PENDAHULUAN

Robot merupakan sebuah peralatan yang dapat melakukan tugas baik dengan kontrol dan pengawasan dari manusia (*remote*), maupun menggunakan program yang sudah ditanamkan sebelumnya (*automatic*) [1]. Seiring perkembangan zaman, robot tidak hanya digunakan di darat saja, tetapi juga di laut (wilayah perairan) dan di udara. Robot yang digunakan di wilayah perairan biasa disebut dengan *Unmanned Surface Vehicle* (USV) [2].

Unmanned Surface Vehicle (USV) merupakan sebuah kapal tanpa awak yang dapat beroperasi di atas permukaan air. USV dapat bekerja secara otomatis ataupun dikendalikan oleh manusia dari jarak yang cukup jauh [3]. Saat ini eksistensi USV sedang mengalami peningkatan di seluruh dunia. Hal ini terjadi karena manfaat potensial dari USV untuk meningkatkan keselamatan dan efisiensi [4]. Selain itu, kapal permukaan otomatis ini juga dapat digunakan sebagai pemantauan (*sampling*) kualitas air [5] dan dimanfaatkan untuk keperluan militer sebagai Alat Utama Sistem Pertahanan (Alutsista) negara [6]. Agar dapat menjalankan tugas tugas tersebut secara otomatis, USV dilengkapi dengan sensor, sistem komunikasi, sistem propulsi, serta sistem deteksi yang membuatnya dapat bernavigasi ketika sedang berlayar [7].

Dalam dunia pelayaran terdapat sarana bantu navigasi yang digunakan untuk menunjang keamanan dan keselamatan kapal ketika berlayar. Salah satu peralatan pembantu navigasi pelayaran adalah pelampung suar (*buoy*) [8]. Berdasarkan peraturan Menteri no.25 tahun 2011 tentang sarana bantu navigasi, *buoy* merupakan sarana bantu navigasi pelayaran apung yang dapat memberikan petunjuk kepada para *navigator* terhadap bahaya atau rintangan dalam navigasi, seperti terumbu karang, perairan dangkal, kerangka kapal, dan menunjukkan perairan aman serta dapat digunakan sebagai tanda batas wilayah suatu negara [9].

Selain memiliki fungsi yang beragam dan krusial dalam dunia navigasi, *buoy* juga memiliki kode warna tertentu yang dapat diindikasikan sebagai peringatan, larangan, perintah, atau petunjuk bagi kapal yang sedang berlayar di area tersebut. Seperti *buoy* warna merah dan *buoy* hijau yang menandakan posisi kiri dan kanan saat kapal akan memasuki pelabuhan [10]. Pada tabel 1. akan dijelaskan mengenai fungsi *buoy* berdasarkan warna yang melekat pada *buoy* tersebut.

Tabel 1. Jenis *Buoy* dan Fungsinya

Tipe *Buoy*	Jenis *Buoy*	Kode Warna Pada *Buoy*	Fungsi
Cardinal	*North Cardinal Buoy*	Hitam di bagian atas dan kuning pada bagian bawah (2 warna)	Menandakan bahwa perairan aman terletak dibagian utara pelampung ini
	East Cardinal Buoy	Hitam di bagian atas, kuning dibagian tengah, dan hitam di bagian bawah (3 warna)	Menandakan bahwa perairan aman terletak dibagian timur dari pelampung ini
	West Cardinal Buoy	Kuning di bagian atas dan bawah pelampung, serta warna kuning di bagian tengah (3 warna)	Menandakan bahwa perairan yang aman terletak di bagian barat dari pelampung tersebut
	South Cardinal Buoy	Kuning di bagian atas dan hitam di bagian bawah (2 warna)	Menandakan perairan yang aman terletak di bagian selatan dari pelampung ini
Lateral	*Starboard Hand*	Hijau	Menunjukkan posisi kanan saat kapal memasuki pelabuhan
	Port Hand	Merah	Menunjukkan posisi kiri saat kapal memasuki pelabuhan

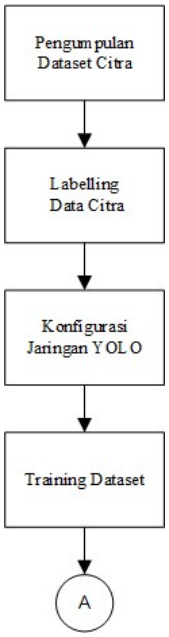
@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya

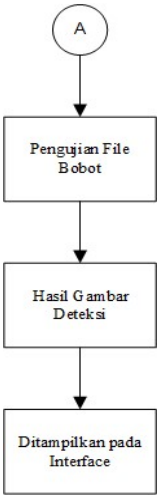
Berdasarkan uraian di atas, maka pendeteksian warna *buoy* merupakan hal yang sangat krusial ketika kapal sedang berlayar, khususnya bagi USV yang bergerak secara otonom dan tidak memiliki awak kapal. Oleh karena itu, diperlukan sistem identifikasi warna yang dapat menentukan dan membedakan warna *buoy* secara cepat, tepat, dan *real-time* yaitu dengan menggunakan metode *You Only Look Once* (YOLO). Metode ini merupakan salah satu pendekatan deteksi objek berbasis *Deep Neural Network* tercanggih dengan performa yang baik pada kecepatan maupun akurasi [11]. Sehingga *Unmanned Surface Vehicle* (USV) akan dapat berlayar dengan baik sesuai peraturan navigasi yang sudah ditentukan.

II. BAHAN DAN METODE

Image processing adalah suatu teknik yang berfungsi untuk memperbaiki, menganalisis, dan mengubah sebuah gambar dalam bentuk 2 dimensi [12]. Terdapat beberapa metode yang bisa digunakan pada proses pengolahan citra (*image proccessing*), seperti *Convolutional Neural Network* (CNN) [13], *Faster* R-CNN [14], dan *Haar Cascade* [15]. Namun pada penelitian ini menggunakan metode *You Only Look Once* (YOLO) untuk mendeteksi warna *buoy*. Metode ini dipilih karena memiliki kecepatan deteksi secara *real-time* hingga 65 *Frame Per Second* (FPS) [16]. Selain itu, algoritma YOLO juga dapat digunakan untuk mendeteksi objek yang statis maupun dinamis [17], [18], sehingga tidak akan mengganggu sistem identifikasi ketika kapal sedang berlayar di laut lepas.

Secara garis besar proses pengolahan citra menggunakan metode *You Only Look Once* adalah sebagai berikut.





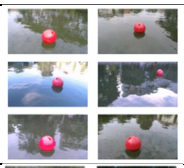
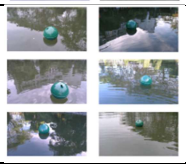
Gambar 1. Proses pengolahan Citra Pada YOLO

Tahapan identifikasi warna *buoy* menggunakan metode YOLO adalah sebagai berikut.

A. Pengumpulan *Dataset* Citra

Dataset merupakan sekumpulan data sekunder yang disusun secara terstruktur dan siap untuk diolah [19]. Pada penelitian ini *dataset* yang digunakan berupa citra atau gambar *buoy* yang memiliki dua klasifikasi warna, yaitu *buoy* merah dan *buoy* hijau. Contoh *dataset* citra *buoy* dapat dilihat pada Tabel 2.

Tabel 2. *Dataset* Citra *Buoy*

Jenis Buoy	Gambar *Dataset*	Waktu	Jumlah	Total
Buoy Merah (1280 x 720 pixel)		Pagi	200	454
		Siang	113	
		Sore	141	
Buoy Hijau (1280 x 720 pixel)		Pagi	180	500
		Siang	136	
		Sore	184	

Total jumlah *dataset* yang digunakan pada penelitian ini adalah 954 citra *buoy* yang diambil dalam berbagai kondisi waktu, yaitu pagi, siang, dan sore hari. Pengambilan dataset dilakukan mulai pukul 07.00 WIB sampai 17.00 WIB. Tujuan dari perbedaan waktu pengambilan *dataset* adalah untuk memperkaya *dataset*, karena perbedaan intensitas cahaya pada ketiga waktu tersebut. Sehingga sistem diharapkan tetap dapat mendeteksi *buoy* meskipun berada di area dengan intensitas cahaya tinggi atau rendah.

B. *Labelling* Data Citra

Tahap pelabelan merupakan proses dimana setiap gambar atau citra yang dikumpulkan pada *dataset* diberikan label sesuai dengan *class* nya masing masing. Proses *labelling* dilakukan dengan cara memberikan kotak atau *bounding box* pada citra benda yang akan dideteksi. Kemudian menentukan *class* dari benda tersebut. Pada penelitian ini, *buoy* merah termasuk dalam *class* 0,

@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya

sedangkan *buoy* hijau termasuk dalam *class* 1. Hasil dari proses *labelling* berupa anotasi gambar yang menampilkan koordinat *width* (X_1 dan X_2) dan *high* (Y_1 dan Y_2) dari *bounding box* menjadi tulisan. Contoh anotasi gambar dari proses *labelling* dapat dilihat pada Tabel 3.

Tabel 3. Anotasi *Dataset*

Class	X_1	Y_1	X_2	Y_2
1	0.921	0.150	0.063	0.093
1	0.844	0.126	0.055	0.083
1	0.786	0.108	0.048	0.069
1	0.740	0.092	0.041	0.067
1	0.701	0.081	0.038	0.061
0	0.146	0.273	0.098	0.150
0	0.260	0.205	0.075	0.116
0	0.362	0.024	0.024	0.046
0	0.386	0.024	0.017	0.034
0	0.428	0.019	0.017	0.030

Tabel 3. Merupakan bentuk hasil anotasi yang berisi titik koordinat *bounding box* dan *class* dari masing masing *buoy*. *Dataset* yang telah diberikan anotasi kemudian diolah menjadi *ground-truth box* dan dibandingkan dengan *predicted box* sehingga dihasilkan *convusion matrix* kemudian dikalkulasikan untuk mendapatkan nilai *Intersection over Union* (IoU), *Precision, Average Precision* (AP), dan *Mean Average Precision* (mAP) [20].

C. Konfigurasi Jaringan YOLO

Konfigurasi jaringan bertujuan untuk pemodelan jaringan yang nantinya akan di*training*. Model metode YOLO yang digunakan pada penelitian ini adalah YOLOv4 dengan konfigurasi seperti pada Tabel 4.

Tabel 4. Konfigurasi Model YOLOV4

Jenis Konfigurasi	Keterangan
Batch	64
Subdivisions	16
Max_batches	4000
Classes	2
Steps	3200, 3600
Filters	21
Width	416
Height	416

Nilai *subdivision* dan *batch* menentukan jumlah pemrosesan gambar dan akan berpengaruh pada hasil *training*. Batas iterasi saat *training* ditentukan dari jumlah *max_batches* yang digunakan. Sehingga ketika iterasi sudah mencapai nilai 4000, maka *training* akan otomatis selesai. Perhitungan nilai *max_batches* dapat dilihat pada persamaan 1.

$$max\ batches = jumlah\ class \text{ x } 2000 \quad (1)$$

Dari nilai *max_batches,* dapat ditentukan nilai steps seperti pada persamaan 2.

$$steps = (80\% \text{ x } max\ batches), \quad (2)$$

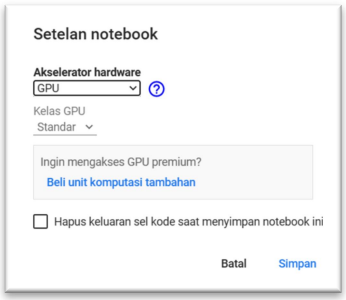
$$(90\ \% \text{ x } max\ batches)$$

Kemudian nilai *filters* di dapatkan dari persamaan 3.

$$filters = (jumlah\ class\ + 5\ ) \text{ x } 3 \quad (3)$$

D. *Training Dataset*

Training dataset dilakukan untuk mendapatkan *file* bobot yang akan diuji. Pada penelitian ini *training* dilakukan dengan menggunakan Google Colab dan dengan bantuan GPU seperti pada Gambar 2.

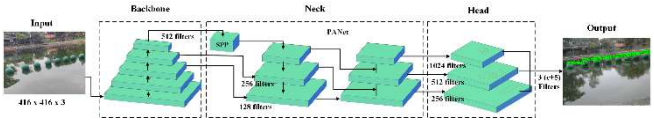


Gambar 2. Training dengan GPU

Graphics Processing Unit atau yang biasa disebut dengan GPU adalah sebuah komponen khusus yang berfungsi untuk mengolah segala hal yang berhubungan dengan grafis. Proses *training* pada penelitian ini menggunakan GPU karena GPU dapat mempercepat proses pelatihan dan komputasi hingga 11 kali lipat dibandingkan ketika menggunakan *Central Processing Unit* atau CPU [21].

E. Arsitektur Jaringan YOLOv4

YOLOv4 merupakan versi modifikasi atau pengembangan dari YOLOv3. Arsitektur YOLOv4 dapat dilihat pada gambar 3.



Gambar 3. Arsitektur YOLOv4

Pada YOLOv4 digunakan *Cross Stage Partial Network* (CSPNet) pada Darknet untuk membuat fitur ekstraksi backbone baru yang disebut dengan CSPDarknet53 [22]. *Backbone* merupakan sebuah model CNN (*Convolutional Neural Network*) yang memiliki banyak layer konvolusi dan *Maxpooling*. Konvolusi merupakan sebuah proses manipulasi citra menggunakan *subwindows* 3 x 3 agar dihasilkan citra baru. Sedangkan *Maxpooling* berfungsi untuk mereduksi input secara spasial menggunakan operasi *downsampling*, yaitu pengambilan nilai terbesar dari operasi tersebut[12].

Setelah melalui beberapa *layer* pada *backbone*. *Output*nya akan disalurkan ke modul penghubung atau *Neck* yang berbeda, termasuk *Spatial Pyramid Pooling* (SPP) dan *Path Aggregation Network* (PANet). Tugas dari SPP dan PANet adalah mengektraksi *feature maps* pada berbagai

@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya

tahap pemrosesan *backbone* dan meningkatkan efisiensi deteksi objek dengan ukuran yang berbeda beda [23].

Tahapan terakhir dari proses deteksi objek menggunakan algoritma YOLO adalah *Head*. *Output* dari *Neck* akan disalurkan ke *Head Network* yang terdiri dari 57 *layer* konvolusi dan 7 *fully connected layer*. *Head Network* bertanggung jawab untuk mengahasilkan *bounding boxes* dan *probability score* untuk setiap objek yang dideteksi.

F. *Convusion Matrix*

Sebelum dilakukan pengujian *dataset* secara *real-time,* perlu dibuat tabel *confusion matrix* untuk mengevaluasi kinerja dari model yang telah dihasilkan setelah proses *training*. Data yang ditampilkan pada tabel *confusion matrix* meliputi jumah prediksi data yang benar (*true positive* dan *true negative*) dan jumlah prediksi data yang salah (*false positive* dan *false negative*) [20].

Confusion matrix dapat digunakan untuk menghitung nilai *accuracy, precision, recall, dan f-score*. *Accuracy* dapat dihitung menggunakan persamaan (4).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+F} \quad (4)$$

Nilai *precision* dapat ditentukan dari persamaan (5).

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Nilai *recall* dapat ditentukan menggunakan persamaan (6).

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

Dan nilai *f-score* dapat dihitung dari persamaan (7).

$$f - score = \frac{Precision \text{ x } recall}{Precision + recall} \quad (7)$$

G. Pengujian Secara *Real-Time*

Setelah proses *training* selesai maka akan dihasilkan *file* bobot dari model YOLO yang digunakan atau yang biasa dikenal dengan *weight file*. pengujian pada penelitian ini akan dilakukan secara *real-time* menggunakan *webcam* dan PC yang telah disiapkan sebelumnya. Spesifikasi PC yang digunakan dapat dilihat pada Tabel 5.

Tabel 5. Spesifikasi PC

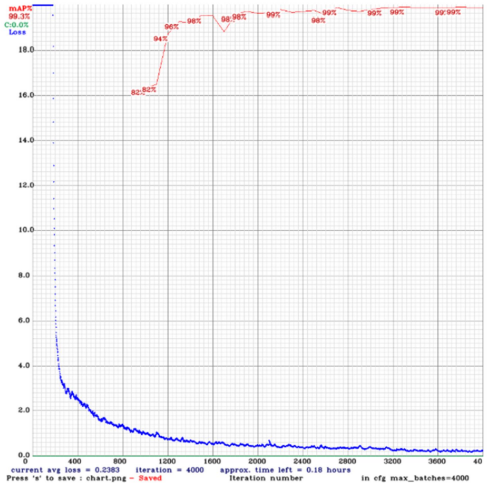
Parameter	***Value***
CPU	*Intel® Core™ i9-9880H Processor (16M Cache, 8 Cores 16 Threads, Base Frequency 2.30 GHz Turbo Frequency 4.80 GHz)*
Memory size	*64 GB*
Memory Types	*2*DDR4 SO-DIMM 260 pin*
Memory Speed	*2666 MHz*
m.2 SSD	*1*M.2 2280 M KEY PCIE3.0*
Sata	*1*SATA3.0 Slot for 2.5inch HDD/SSD*
Processor Graphics	*Intel® UHD Graphics 630 + NVIDIA GeForce GTX 1650 4GB GDDR5 2 GPU*
Adapter Output	*DC 19V/7.89A*

Spesifikasi PC yang digunakan akan berpengaruh kepada nilai *Frame Per Second* (FPS) yang dihasilkan. Semakin tinggi spesifikasi yang digunakan maka nilai FPS juga akan semakin tinggi.

III. Hasil dan Pembahasan

A. Hasil *Training Dataset*

Setelah dilakukan proses *training dataset* selama kurang lebih 10 jam, maka dihasilkan grafik hasil *training* seperti pada Gambar 4.



Gambar 4. Grafik Hasil *Training Dataset*

Berdasarkan grafik hasil *training dataset* pada Gambar 4. Didapatkan nilai *Mean Average Precision* (maP) sebesar 99,3%. Semakin tinggi niai maP pada hasil *training* YOLO, maka kinerja model tersebut akan semakin baik dalam proses deteksi objek. Selain itu dari grafik juga ditunjukkan nilai *average loss* yaitu sebesar 0,2383. Pada metode YOLO *average loss* mengacu pada rata rata dari kesalahan prediksi model selama pelatihan berlangsung. Sehingga semakin kecil nilai *average loss* yang dihasilkan dari proses *training*, maka kinerja model dalam mendeteksi objek akan semakin baik.

B. Hasil Pengujian Model

Model diuji secara manual dengan memasukan gambar *dataset* satu per satu, sehingga didapatkan nilai *confusion matrix* seperti pada Tabel 6.

Tabel 6. Hasil Pengujian Model

Load Model		**YOLOv4**
Bouy Merah	TP	477
	TN	0
	FP	15
	FN	8
	Accuracy	0.954
	Precision	0.969
	Recall	0.983
Bouy Hijau	TP	438
	TN	0
	FP	7
	FN	9
	Accuracy	0.964

@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya

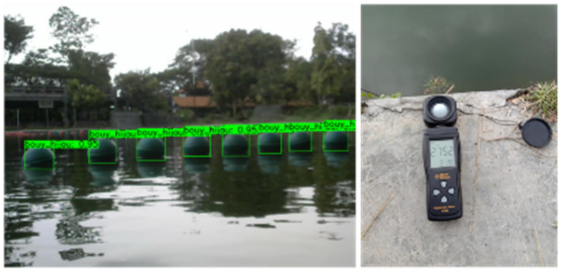
Load Model		YOLOv4
	Precision	0.984
	Recall	0.979

Dapat dilihat pada Tabel 6. Bahwa Nilai akurasi pendeteksian *buoy* merah sebesar 95% dan *buoy* hijau sebesar 96% (nilai diubah menjadi persentase). Sehingga tingkat keberhasilan model dalam melakukan prediksi yang benar pada *dataset* dapat dikatakan cukup baik.

C. Hasil Pengujian Secara *Real-Time*

Pengujian dilakukan secara *real-time* pada kolam uji, dimana jarak terdekat antara kamera dengan *buoy* adalah 3 meter dan jarak terjauhnya adalah 5 meter. Perangkat yang digunakan pada saat pengujian adalah *webcam, lux meter* yang digunakan sebagai alat ukur intensitas cahaya, dan PC sesuai dengan spesifikasi yang telah disebutkan pada Tabel 5. Berikut merupakan hasil pengujian model terhadap *buoy* merah dan *buoy* hijau dalam kondisi waktu dan intensitas cahaya yang berbeda beda.

Gambar 5. Hasil Deteksi *Buoy* Merah Pada 15.940 lux



Gambar 6. Hasil Deteksi *Buoy* Hijau pada 2.752 lux

Gambar 5. dan 6. menyajikan hasil deteksi *buoy* merah dan hijau pada intensitas cahaya yang berbeda. Pengukuran intensitas cahaya dilakukan dengan menggunakan lux meter yang telah dikalibrasi sebelumnya. Terlihat pada gambar seluruh *buoy* terdeteksi dengan baik. Dimana jumlah *buoy* yang berderet adalah 8 buah dan *buoy* yang terdeteksi juga ada 8 buah. Peneliti juga melakukan pengujian secara acak mulai dari pukul 08.00 WIB sampai pukul 17.00 WIB. Tujuannnya adalah untuk memastikan tingkat keberhasilan algoritma YOLO dalam mendeteksi objek, karena intensitas cahaya pada rentang waktu tersebut jelas berbeda beda. Hasil pengujian deteksi *buoy* merah dan hijau dapat dilihat pada Tabel 7. dan 8.

Tabel 7. Hasil Deteksi *Buoy* Merah

No.	Pukul (WIB)	Intensitas Cahaya (lux)	Jumlah Buoy		Persen Deteksi (%)
			Sebenarnya	Terdeteksi	
1	08.32	15.940	8	8	100
2	08.52	23.490	8	8	100
3	09.16	40.110	8	8	100
4	09.46	44.020	8	8	100
5	10.08	46.140	8	8	100
6	10.38	44.960	8	8	100
7	10.54	45.000	8	7	87,5
8	12.27	63.350	8	7	87,5
9	12.47	38.260	8	8	100
10	13.00	92.630	8	8	100
11	13.24	72.250	8	8	100
12	13.52	53.760	8	8	100
13	14.18	32.980	8	8	100
14	14.32	28.530	8	8	100
15	15.10	21.130	8	8	100
16	15.31	23.650	8	8	100
17	15.49	13.190	8	8	100
18	16.04	12.930	8	8	100
19	16.26	9.738	8	8	100
20	16.32	6.126	8	8	100
21	16.57	2.752	8	8	100

Tabel 8. Hasil Deteksi *Buoy* Hijau

No.	Pukul (wib)	Intensitas Cahaya (lux)	Jumlah Buoy		Persen Deteksi (%)
			Sebenarnya	Terdeteksi	
1	08.32	15.940	8	8	100
2	08.52	23.490	8	8	100
3	09.16	40.110	8	8	100
4	09.46	44.020	8	8	100
5	10.08	46.140	8	8	100
6	10.38	44.960	8	8	100
7	10.54	45.000	8	8	100
8	12.27	63.350	8	8	100
9	12.47	38.260	8	8	100
10	13.00	92.630	8	8	100
11	13.24	72.250	8	8	100
12	13.52	53.760	8	8	100
13	14.18	32.980	8	8	100
14	14.32	28.530	8	8	100
15	15.10	21.130	8	8	100
16	15.31	23.650	8	8	100
17	15.49	13.190	8	8	100
18	16.04	12.930	8	8	100
19	16.26	9.738	8	8	100
20	16.32	6.126	8	8	100

@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya

No.	Pukul (wib)	Intensitas Cahaya (lux)	Jumlah Buoy		Persen Deteksi (%)
			Sebenarnya	Terdeteksi	
21	16.57	2.752	8	8	100

Berdasarkan Tabel 7. terdapat 2 kesalahan pendeteksian jumlah *buoy* merah yang terjadi pada intensitas cahaya 45.000 lux dan 63.350 lux. Kesalahan deteksi tersebut bisa saja terjadi karena cahaya matahari yang terlalu terang, sehingga membuat *buoy* merah menjadi silau dan tidak terdeteksi. Sedangkan pada Tabel 8. seluruh *buoy* hijau terdeteksi dengan baik. Hal ini membuktikan bahwa algoritma YOLO tetap dapat mendeteksi warna dari objek dengan tepat, meskipun berada pada intensitas cahaya yang berbeda. Namun ketika cahaya terlalu gelap atau terlalu terang maka nilai akurasi dari deteksi benda akan berkurang.

IV. KESIMPULAN

Setelah dilakukan *training dataset* dan pengujian identifikasi warna *buoy* menggunakan metode YOLO secara *real-time*, didapatkan nilai *mean average precision* (maP) sebesar 99,3% Dan nilai *average loss* sebesar 0,2383. Selain itu, hasil pendeteksian warna objek menunjukkan nilai yang baik. Pada *buoy* merah rata rata nilai deteksi sebesar 98,8 % dan rata rata nilai deteksi pada *buoy* hijau sebesar 100 %. yang menandakan bahwa algoritma YOLO dapat diimplementasikan untuk mendeteksi objek yang sama dengan warna yang berbeda pada wilayah dengan intensitas cahaya tinggi maupun rendah.

Penelitian ini dapat digunakan sebagai refrensi untuk penelitian selanjutnya. Namun terdapat beberapa hal yang perlu diperhatikan, seperti memperbanyak jumlah *dataset* yang digunakan pada masing masing *class* dan kondisi. Tujuannya agar hasil akurasi deteksi menjadi lebih baik. Selain itu, spesifikasi pc yang digunakan juga perlu diperhatikan supaya proses *training* dan *running* program deteksi bisa lebih cepat dan maksimal.

DAFTAR PUSTAKA

[1] N. L. Husni, S. Rasyad, M. S. Putra, Y. Hasan, and J. Al Rasyid, "Pengaplikasian sensor warna pada navigasi line traking robot sampah," *Ampere*, vol. 4, no. 2, pp. 297–306, 2019.

[2] N. F. Satria, S. Kuswadi, and N. Arifin, "Rancang bangun unmanned surface vehicle berbasis adaptive morphology di air dan darat," *Semin. Nas. Terap. Ris. Inov.*, vol. 6, no. 1, pp. 379–387, 2020.

[3] D. Ariateja, U. D. Fatmawati, and I. A. Dahlan, "Instrumentasi pemantauan perairan berbasis telemetri pada prototipe unmanned surface vehicle (USV)," *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 7, no. 2, p. 200, 2021, doi: 10.24036/jtev.v7i2.113096.

[4] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles I: a review," *J. Mar. Sci. Technol.*, vol. 26, no. 4, pp. 1292–1306, 2021, doi: 10.1007/s00773-020-00787-6.

[5] L. Steccanella, D. D. Bloisi, A. Castellini, and A. Farinelli, "Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring," *Rob. Auton. Syst.*, vol. 124, 2020, doi: 10.1016/j.robot.2019.103346.

[6] H. Priyono, S. Aritong, and M. Akbar, "Pengembangan teknologi kapal tanpa awak guna mendukung operasi dan latihan tni angkatan laut," *J. Indones. Sos. Sains*, vol. 3, no. 3, pp. 527–537, 2022, doi: 10.36418/jiss.v3i3.555.

[7] D. Permana, M. Rivai, and N. Irfansyah, "30999-83886-1-PB.pdf," vol. 7, no. 2, 2018.

[8] E. T. Wahyuni, "Peranan sarana bantu navigasi pelayaran terhadap keselamatan pelayaran," *Natl. Semin. Marit. Interdiscip. Stud. 1*, vol. 1 no 1, pp. 269–274, 2019.

[9] R. Theresia, "Prosedur darurat penanganan kapal kandas terhadap lct . total iii di area buoy 14 dan buoy 16 menuju sungai lais di perairan mahakam," vol. 12, no. 1, pp. 34–41, 2022.

[10] Surnata, N. Hayatun, K. Alam, and E. Agustini, "Semiotika rambu-rambu lalu lintas laut elfita agustini abstrak semiotics of marine traffic signs abstract dalam peraturan menteri perhubungan republik indonesia nomor 13 tahun 2014 tentang lalu lintas bab 1 pasal 1 bahwa rambu lalu lintas adalah bagian," vol. 4, no. 2, pp. 443–456, 2021.

[11] W. Fang, L. Wang, and P. Ren, "Tinier-yolo: a real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020, doi: 10.1109/ACCESS.2019.2961959.

[12] A. Khumaidi and R. L. Pradana, "Identifikasi penyebab cacat pada hasil pengelasan dengan image processing menggunakan metode yolo," *J. Tek. Elektro dan Komput. TRIAC*, vol. 9, no. 3, pp. 107–112, 2022, [Online]. Available: https://journal.trunojoyo.ac.id/triac/article/view/15997

[13] V. M. P. Salawazo, D. P. J. Gea, R. F. Gea, and F. Azmi, "Implementasi metode convolutional neural network (cnn) pada peneganalan objek video cctv," *J. Mantik Penusa*, vol. 3, no. 1, pp. 74–79, 2019.

[14] M. F. Rahman and B. Bambang, "Deteksi sampah pada real-time video menggunakan metode faster r-cnn," *Appl. Technol. Comput. Sci. J.*, vol. 3, no. 2, pp. 117–125, 2021, doi: 10.33086/atcsj.v3i2.1846.

[15] A. Thariq and R. Y. Bakti, "Sistem deteksi masker dengan metode haar cascade pada era new normal covid-19," *J. Sist. dan Teknol. Inf.*, vol. 9, no. 2, p. 241, 2021, doi: 10.26418/justin.v9i2.44309.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, [Online]. Available: http://arxiv.org/abs/2004.10934

[17] M. Shen *et al.*, "A deep learning based automatic defect analysis framework for in-situ tem ion irradiations," *Comput. Mater. Sci.*, vol. 197, no. May, p. 110560, 2021, doi: 10.1016/j.commatsci.2021.110560.

[18] J. Han, Y. Liao, J. Zhang, S. Wang, and S. Li, "Target fusion detection of lidar and camera based on the improved yolo algorithm," *Mathematics*, vol. 6, no. 10, 2018, doi: 10.3390/math6100213.

[19] R. T. Prasetio and E. Ripandi, "Optimasi klasifikasi jenis hutan menggunakan deep learning berbasis optimize selection," *J. Inform.*, vol. 6, no. 1, pp. 100–106, 2019, doi: 10.31311/ji.v6i1.5176.

[20] K. A. Shianto, K. Gunadi, and E. Setyati, "Deteksi jenis mobil menggunakan metode yolo dan faster r-cnn," *J. Infra*, vol. 7, no. 1, pp. 157–163, 2019, [Online]. Available: http://publication.petra.ac.id/index.php/teknik-informatika/article/view/8065

[21] A. Thohari and G. B. Hertantyo, "Implementasi convolutional neural network untuk klasifikasi pembalap motogp berbasis gpu," *Proc. Conf. Electr. Eng. Telemat. Ind. Technol. Creat. Media*, pp. 50–55,

@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya

2018.

[22] U. Nepal and H. Eslamiat, “Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs,” *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020464.

[23] K. Roszyk, M. R. Nowicki, and P. Skrzypczyński, “Adopting the yolov4 architecture for low-latency multispectral pedestrian detection in autonomous driving,” *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22031082.

@ 2023 Faiz Romadloni, Joko Endrasmono, Zindhu Maulana Ahmad Putra, Agus Khumaidi, Isa Rachman, Ryan Yudha Adhitya