



PPNS POLITEKNIK
PERKAPALAN
NEGERI SURABAYA

TUGAS AKHIR(AE43250)

**IMPLEMENTASI METODE YOLOv8 UNTUK DETEKSI OBJEK
LINTASAN *AUTONOMOUS UNDERWATER ROBOT (AUR)*
PADA *EVENT SAUVC***

Erutan Dwi Sajiwo
NRP. 0921040008

Dosen Pembimbing
Ryan Yudha Adhitya, S.ST., M.T
Zindhu Maulana Ahmad Putra., S.ST., M.Tr.T.

PROGRAM STUDI D4 TEKNIK OTOMASI
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2025

(Halaman ini sengaja dikosongkan)



PPNS POLITEKNIK
PERKAPALAN
NEGERI SURABAYA

TUGAS AKHIR(AE43250)

**IMPLEMENTASI METODE YOLOv8 UNTUK DETEKSI OBJEK
LINTASAN *AUTONOMOUS UNDERWATER ROBOT (AUR)*
PADA *EVENT SAUVC***

Erutan Dwi Sajiwo
NRP. 0921040008

Dosen Pembimbing
Ryan Yudha Adhitya, S.ST., M.T
Zindhu Maulana Ahmad Putra., S.ST., M.Tr.T.

PROGRAM STUDI D4 TEKNIK OTOMASI
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2025

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TUGAS AKHIR

IMPLEMENTASI METODE YOLOv8 UNTUK DETEKSI OBJEK LINTASAN *AUTONOMOUS UNDERWATER ROBOT* (AUR) PADA *EVENT* SAUVC

Disusun Oleh:
Erutan Dwi Sajiwo
0921040008

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan
Program Studi Teknik Otomasi
Jurusan Teknik Kelistrikan Kapal
POLITEKNIK PERKAPALAN NEGERI SURABAYA

Disetujui oleh Tim penguji Tugas Akhir Tanggal Ujian : 17 Juli 2025
Periode Wisuda : Oktober 2025

Menyetujui,

Dosen Penguji

1. Muhammad Khairul Hasin, S.Kem, M.Kom.
2. Dimas Pristovani Riananda, S.ST., M.T.
3. Adianto, S.T., M.T.
4. Ryan Yudha Adhitya, S.ST., M.T.

NUPTK

(9557766667130313)
(9263771672130193)
(1034755656130133)
(9948769670130352)

Tanda Tangan

(.....)
(.....)
(.....)
(.....)

Dosen Pembimbing

1. Ryan Yudha Adhitya, S.ST., M.T.
2. Zindhu Maulana Ahmad Putra, S.ST., M.Tr.T.

NUPTK

(9948769670130352)
(7255772673130273)

Tanda Tangan

(.....)
(.....)

Menyetujui
Ketua Jurusan,



Isa Rachman, S.T., M.T.

NIP. 198008162008121001


Mengetahui
Koordinator Program Studi,

Agus Khumaidi, S.ST., M.T.

NIP. 199308172020121004

(Halaman ini sengaja dikosongkan)

LEMBAR BEBAS PLAGIAT

 PPNS	<u>PERNYATAAN BEBAS PLAGIAT</u>	No. : F.WD I. 021 Date : 3 Nopember 2015 Rev. : 01 Page : 1 dari 1
---	--	--

Yang bertandatangan dibawah ini :

Nama : Erutan Dwi Sajiwo

NRP. : 0921040008

Jurusan/Prodi : Teknik Kelistrikan Kapal/D4 Teknik Otomasi

Dengan ini menyatakan dengan sesungguhnya bahwa :

Tugas Akhir yang akan saya kerjakan dengan judul :

**IMPLEMENTASI METODE YOLOv8 UNTUK DETEKSI OBJEK LINTASAN
AUTONOMOUS UNDERWATER ROBOT (AUR) PADA EVENT SAUVC**

Adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain.

Apabila dikemudian hari terbukti terdapat plagiat dalam karya ilmiah tersebut,
maka saya bersedia menerima **sanksi** sesuai ketentuan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 27 Agustus 2025

Yang membuat pernyataan,



(Erutan Dwi Sajiwo)
NRP. 0921040008

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Alhamdulillahirabbilalamin, puji syukur atas ke hadirat allah ALLAH Subhanahu wa Ta'ala yang telah memberikan segala nikmat serta limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “ **Implementasi Metode YOLOv8 Untuk Deteksi Objek *Lintasan Autonomous Underwater Robot (AUR)* Pada Event SAUVC** ” dengan baik. Sholawat serta salam mudah mudahan tetap tercurahkan kepada Nabi Muhammad Rasulullah Shallallahu ‘alaihi wasallam yang senantiasa membimbing kita dari kejahiliyahan menuju agama islam.

Tugas Akhir ini disusun oleh penulis sebagai salah satu persyaratan untuk menyelesaikan Pendidikan Program Studi D4 Teknik Otomasi dan mendapatkan gelar Sarjana Terapan di Politeknik Perkapalan Negeri Surabaya. Penulis sangat menyadari bahwa keberhasilan pengerjaan Tugas Akhir ini tidak lepas dari bimbingan, dukungan, serta motivasi dan bantuan dari berbagai pihak. Oleh karena itu, melalui kesempatan ini penulis ingin mengucapkan terima kasih khususnya kepada :

1. Orang tua penulis, Bapak Muri dan Ibu Yuni Wulandari serta kakak Violita Pita Nugraheni yang selalu mendoakan, memberikan dukungan, dan mengingatkan penulis untuk tetap semangat dalam mengerjakan tugas akhir ini hingga selesai.
2. Bapak Rachmad Tri Soelistijono, ST., M.T. selaku Direktur Politeknik Perkapalan Negeri Surabaya.
3. Bapak Isa Rachman, S.T., M.T. selaku Ketua Jurusan Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya.
4. Bapak Agus Khumaidi, S.ST., M.T. selaku Ketua Program Studi Teknik Otomasi Politeknik Perkapalan Negeri Surabaya.
5. Bapak Ryan Yudha Adhitya, S.ST., M.T. selaku Dosen Pembimbing I yang telah memberikan solusi dan arahan kepada penulis dalam menyelesaikan Tugas Akhir ini dengan baik.

6. Bapak Zindhu Maulana Ahmad Putra, S.ST., M.Tr.T. selaku dosen pembimbing II yang telah sabar membimbing, memberikan dukungan, memberikan Solusi, serta mengajarkan banyak ilmu sehingga Tugas Akhir ini dapat terselesaikan dengan baik.
7. Seluruh Dosen Jurusan Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya yang telah memberikan ilmu dan pengetahuannya selama penulis menjalankan studi di PPNS.
8. Seluruh keluarga besar Teknik Otomasi khususnya Angkatan 2021 atas 4 tahun yang berkesan.
9. Seluruh keluarga besar UKM (*Unit Kegiatan Mahasiswa*) MRC (*Marine Robotic Community*) yang telah memfasilitasi penulis untuk belajar dan berkembang sehingga penulis bisa mendapatkan ilmu dan pengalaman yang berharga dalam proses penyelesaian Tugas Akhir ini.
10. Teman-teman AEROships dan HYDROships yang sudah banyak membantu penulis dalam menyelesaikan tugas akhir ini.
11. Serta semua pihak yang tidak dapat disebutkan satu persatu. Semoga kebbaikannya dibalas oleh ALLAH Subhanahu wa Ta'ala dengan berlipat ganda, amin.

Penulis menyadari bahwa dalam dalam pengerjaan dan penulisan Tugas Akhir ini masih banyak kekurangan. Sehingga kritik dan saran sangat dibutuhkan untuk penyempurnaan Tugas Akhir ini. Penulis juga berharap semoga Tugas Akhir ini dapat bermanfaat dan berguna bagi penulis sendiri dan bagi pembacanya.

Surabaya, 08 Agustus 2025

Penulis,

Erutan Dwi Sajiwo

IMPLEMENTASI METODE YOLOv8 UNTUK DETEKSI OBJEK LINTASAN *AUTONOMOUS UNDERWATER ROBOT* (AUR) PADA *EVENT SAUVC*

Erutan Dwi Sajiwo

ABSTRAK

Penelitian ini mengimplementasikan metode YOLOv8 untuk deteksi objek pada lintasan Autonomous Underwater Robot (AUR) dalam kompetisi Singapore Autonomous Underwater Vehicle Challenge (SAUVC) 2025. AUR dirancang untuk mendeteksi objek berwarna, seperti gate, flare (orange, kuning, merah, biru), dan drum biru, guna menjalankan misi otonom termasuk navigasi, akuisisi target, reakuisi target, serta komunikasi dan lokalisasi. Dataset dikumpulkan sebanyak 5.175 untuk pelatihan dan 2.075 untuk validasi, terdiri dari 8 kelas, dengan hasil pelatihan model YOLOv8n selama 150 epoch mencapai recall rata-rata 91,45%, akurasi 51,6%, presisi 20,8%, dan F1-score 32,5%. Ketidakseimbangan dataset antarkelas menyebabkan performa tidak optimal, sehingga diperlukan peningkatan variasi data. Integrasi sensor HWT905 Attitude Angle Sensor memastikan keseimbangan AUR dengan akurasi rata-rata 99,42% dan error 0,78% berdasarkan pengujian sepuluh kali menggunakan busur derajat. Pengujian real-time di kolam Atlas Sport (14x5,5x2,5 m) pada malam hari dengan intensitas cahaya 102 lux menunjukkan akurasi deteksi rata-rata 86,64% dan error 13,36% dari sepuluh percobaan, memungkinkan AUR menghindari flare orange, melewati gate, menabrak flare secara berurutan, serta menjatuhkan dan mengambil bola pada drum biru sesuai aturan SAUVC. Penelitian ini memperkuat pengembangan teknologi AUR yang andal untuk lingkungan bawah air yang kompleks.

Kata kunci : YOLOv8, AUR, SAUVC, Deteksi Objek, Pengujian *Real-time*.

(Halaman ini sengaja dikosongkan)

IMPLEMENTATION OF THE YOLOV8 METHOD FOR OBJECT DETECTION IN THE PATH OF AUTONOMOUS UNDERWATER ROBOTS (AUR) AT THE SAUVC EVENT

Erutan Dwi Sajiwo

ABSTRACT

This study implements the YOLOv8 method for object detection in the trajectory of an Autonomous Underwater Robot (AUR) during the Singapore Autonomous Underwater Vehicle Challenge (SAUVC) 2025. The AUR is designed to detect colored objects, such as gates, flares (orange, yellow, red, blue), and blue drums, to execute autonomous missions, including navigation, target acquisition, target reacquisition, and communication and localization. A dataset comprising 5,175 training images and 2,075 validation images, covering eight classes, was used to train the YOLOv8n model over 150 epochs, achieving an average recall of 91.45%, accuracy of 51.6%, precision of 20.8%, and F1-score of 32.5%. Class imbalance in the dataset led to suboptimal performance, necessitating improved data variety. Integration of the HWT905 Attitude Angle Sensor ensured AUR stability, with an average accuracy of 99.42% and an error of 0.78% across ten trials using a protractor. Real-time testing in the Atlas Sport pool (14x5.5x2.5 m) at night, with a light intensity of 102 lux, demonstrated an average detection accuracy of 86.64% and an error rate of 13.36% over ten trials, enabling the AUR to avoid orange flares, pass through gates, sequentially collide with flares, and drop and retrieve a ball at the blue drum in accordance with SAUVC rules. This research enhances the development of reliable AUR technology for complex underwater environments..

Keywords: YOLOv8, AUR, SAUVC, Object Detection, Real-time Testing.

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
PERNYATAAN BEBAS PLAGIAT	v
KATA PENGANTAR.....	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL	xvii
DAFTAR GAMBAR.....	xix
DAFTAR NOTASI.....	xxiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	4
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	5
BAB 2 TINJAUAN PUSTAKA	7
2.1 Kajian Penelitian Terdahulu	7
2.2 Pengolahan Citra	7
2.3 <i>Pixel, Resolusi dan Intensitas</i>	8
2.4 Model Warna RGB.....	9
2.5 <i>You Only Look Once</i> (YOLO).....	10
2.6 Software Yang Digunakan.....	13
2.6.1 Python.....	13
2.6.2 <i>Compute Unified Device Architecture</i> (CUDA).....	14
2.6.3 <i>Open Computer Vision</i> (OpenCV)	14
2.6.4 TensorRT	15
2.6.5 Arduino IDE	16
2.6.6 Visual Studio Code.....	17
2.7 Hardware Yang Digunakan	17

2.7.1	<i>Webcam</i>	17
2.7.2	<i>Pressure Sensor</i>	18
2.7.3	<i>Attitude Angle Sensor</i>	19
2.7.4	<i>Buck Converter</i>	20
2.7.5	Mini PC	21
2.7.6	ESP32	22
2.7.7	<i>Electronic Speed Controller (ESC)</i>	24
2.7.8	<i>Motor Thruster</i>	24
2.7.9	BLDC Motor	25
2.7.10	<i>Push Pull Solenoid</i>	26
2.7.11	Modul <i>Interface</i>	27
2.7.12	Relay	27
2.7.13	<i>Switch</i>	29
2.7.14	Baterai <i>Li-ion</i>	30
2.8	Komunikasi <i>Serial</i>	32
2.8.1	Komunikasi <i>Serial Sinkron</i>	33
2.8.2	Komunikasi <i>Serial Asinkron</i>	33
BAB 3	METODOLOGI PENELITIAN	35
3.1	Konsep Penelitian	35
3.1.1	<i>State Of The Art</i> (Keterbaruan dari penelitian)	35
3.1.2	Prinsip Kerja Sistem	36
3.1.3	Sistem Deteksi Objek Dengan Metode YOLOv8	37
3.2	Diagram Alur Penelitian	40
3.3	Identifikasi Awal	40
3.3.1	Identifikasi Masalah	40
3.3.2	Penetapan Tujuan dan Manfaat Penelitian	41
3.3.3	Studi Literatur	41
3.3.4	Analisa Kebutuhan Sistem	41
3.4	Desain dan Perencanaan Sistem	42
3.4.1	Desain Sistem Mekanik	42
3.4.2	Perencanaan <i>Hardware</i>	45
3.4.3	Desain Lintasan	46

3.5	<i>Dataset</i>	47
3.5.1	Penerapan Rumus Algoritma YOLO.....	49
BAB 4	HASIL DAN PEMBAHASAN	51
4.1	Pengujian Sensor dan Aktuator	51
4.1.1	Pengujian Sensor <i>Webcam</i>	51
4.1.2	Pengujian Sensor <i>Switch</i>	52
4.1.3	Pengujian <i>Pressure</i> Sensor	53
4.1.4	Pengujian <i>Attitude Angle</i> Sensor	54
4.1.5	Pengujian Aktuator Motor <i>Thruster</i>	57
4.1.6	Pengujian Aktuator BLDC Motor	58
4.1.7	Pengujian Aktuator <i>Push Pull Solenoid</i>	59
4.2	Pembuatan Sistem Mekanik	60
4.2.1	Pembuatan Kerangka AUR	60
4.2.2	Pembuatan Ballast	61
4.3	Pembuatan dan Perakitan Sistem Elektrik.....	62
4.3.1	Pembuatan dan Perakitan Sistem Mainboard	62
4.3.2	Instalasi Sistem Elektrik pada AUR	64
4.3.3	Perakitan Seluruh Bagian AUR.....	65
4.4	Pembuatan Dataset	67
4.4.1	Pengumpulan <i>dataset</i>	67
4.4.2	<i>Labelling Dataset</i>	69
4.4.3	<i>Split Dataset</i>	70
4.5	Training Model YOLOv8.....	71
4.5.1	Pengaturan Parameter <i>Training</i>	71
4.5.2	Proses <i>Training</i>	72
4.5.3	Hasil <i>Training</i>	74
4.5.4	Hasil Pengujian menggunakan Confusion Matrix	75
4.6	Deteksi Objek Secara <i>Real-time</i>	88
BAB 5	KESIMPULAN DAN SARAN	102
5.1	Kesimpulan.....	103
5.2	Saran	104
	DAFTAR PUSTAKA	105

LAMPIRAN.....	109
Lampiran 1 Sertifikat Juara 2 Internasional SAUVC.....	109
Lampiran 2 Biodata Mahasiswa	111

DAFTAR TABEL

Tabel 2. 1 Spesifikasi Mini PC Jetson Orin NX	21
Tabel 2. 2 Spesifikasi distribusi penggunaan Baterai Li-ion	31
Tabel 3. 1 Jumlah Kebutuhan Sistem.....	42
Tabel 4. 1 Data Hasil Pengujian <i>Pressure</i> Sensor MS5837-30BA.....	54
Tabel 4. 2 Data Hasil Pengujian Sumbu X HWT905	55
Tabel 4. 3 Data Hasil Pengujian Sumbu Y HWT905	55
Tabel 4. 4 Data Hasil Pengujian Sumbu Z HWT905	56
Tabel 4. 5 Data Pengukuran Kecepatan <i>Motor Thruster</i> Menggunakan <i>Tachometer</i>	57
Tabel 4. 6 Data Pengukuran Kecepatan BLDC Motor Menggunakan <i>Tachometer</i>	58
Tabel 4. 7 Hasil Pengambilan Data Objek Lintasan SAUVC.....	68
Tabel 4. 8 Parameter Training.....	72
Tabel 4. 9 Hasil Pengujian Model dengan <i>Confusion Matrix</i>	85
Tabel 4. 10 Tabel 4. 10 Hasil Pengujian Deteksi Secara Langsung	89

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2. 1 Koordinat <i>pixel</i>	9
Gambar 2. 2 Model Gambar RGB	10
Gambar 2. 3 Deteksi Objek YOLO	10
Gambar 2. 4 Asitektur YOLOv8	12
Gambar 2. 5 Perbandingan <i>YOLOv8</i> dengan metode deteksi lainnya	12
Gambar 2. 6 Logo Python	13
Gambar 2. 7 Logo NVidia CUDA	14
Gambar 2. 8 Logo OpenCV	15
Gambar 2. 9 Logo TensorRT	15
Gambar 2. 10 Tampilan <i>Software</i> Arduino IDE	16
Gambar 2. 11 Tampilan <i>Software</i> VS Code	17
Gambar 2. 12 <i>Webcam</i> Logitech C930E	18
Gambar 2. 13 <i>Pressure</i> Sensor MS5837-30BA	18
Gambar 2. 14 HWT905 <i>Attitude Angle</i> Sensor	19
Gambar 2. 15 Modul XL4015 (<i>DC to DC Buck Converter</i>)	20
Gambar 2. 16 Modul MP1584 (<i>DC to DC Buck Converter</i>)	20
Gambar 2. 17 Mini PC NVidia Jetson Orin NX	22
Gambar 2. 18 <i>Microcontroller</i> ESP32 Wroom 32U	23
Gambar 2. 19 ESC ROV MAKER	24
Gambar 2. 20 T200 <i>Brushless</i> ROV <i>Thruster</i>	25
Gambar 2. 21 BLDC Motor F2429	25
Gambar 2. 22 <i>Push Pull Solenoid</i> JF-0826B	26
Gambar 2. 23 <i>Tether Interface Board</i>	27
Gambar 2. 24 Relay Sngle SRD-05VDC-SL-C	28
Gambar 2. 25 Relay WM686 RL/180-12	28
Gambar 2. 26 <i>Toggle Switch</i> MTS-102	29
Gambar 2. 27 <i>Magnetic Switch</i> MC-38	30
Gambar 2. 28 M10 <i>Underwater Switch</i>	30
Gambar 2. 29 Li-ion Battery SONY VTC6 18650	31

Gambar 3. 1 Diagram Blok Sistem	36
Gambar 3. 2 Diagram Alur Proses Deteksi Objek	37
Gambar 3. 3 <i>Matrix</i> Citra Digital 4x4	38
Gambar 3. 4 Ilustrasi Proses Konvolusi Citra	38
Gambar 3. 5 Hasil Proses Konvolusi Citra	39
Gambar 3. 6 Hasil Proses Max Pooling Citra	39
Gambar 3. 7 Alur Penelitian.....	40
Gambar 3. 8 Desain AUR Tampak Isometri	43
Gambar 3. 9 Desain AUR Tampak Atas	43
Gambar 3. 10 Desain AUR Tampak Belakang	44
Gambar 3. 11 Desain AUR Tampak Depan.....	44
Gambar 3. 12 Desain AUR Tampak Samping.....	45
Gambar 3. 13 Perancangan Hardware.....	45
Gambar 3. 14 Desain Lintasan SAUVC 2025	47
Gambar 3. 15 FFlowchart Proses Pembuatan <i>Dataset</i>	48
Gambar 4. 1 a) Resolusi Citra Hasil Tangkapan <i>Webcam Logitech C390e</i> b) Indikator <i>Webcam</i> Terhubung Dengan PC atau Laptop	51
Gambar 4. 2 a) Kondisi <i>Magnetic Switch MC-38 Off</i> b) Kondisi <i>Magnetic Switch MC-38 On</i>	52
Gambar 4. 3 a) Kondisi <i>M10 Underwater Switch</i> Sedang Tidak Ditekan b) Kondisi <i>M10 Underwater Switch</i> Sedang Ditekan	52
Gambar 4. 4 a) Kondisi <i>M10 Toggle Switch MTS-102</i> Sedang Tidak Ditekan b) Kondisi <i>M10 Toggle Switch MTS-102</i> Sedang Ditekan.....	53
Gambar 4. 5 Pengujian <i>Pressure Sensor MS5837-30BA</i>	54
Gambar 4. 6 Perhitungan Sudut <i>HWT905 Attitude Angle Sensor</i> dengan Busur .	55
Gambar 4. 7 Pengujian Kecepatan <i>Motor Thruster</i> Menggunakan <i>Tachometer</i> ..	57
Gambar 4. 8 Proses Pengukuran Kecepatan BLDC Motor Menggunakan Tachometer.....	58
Gambar 4. 9 a) Kondisi <i>Push Pull Solenoid JF-0826B</i> Tidak Menerima Tegangan 12V b) Kondisi <i>Push Pull Solenoid JF-0826B</i> Menerima Tegangan 12V....	59
Gambar 4. 10 Hasil CNC Rangka AUR	60

Gambar 4. 11 Hasil Penyatuan Rangka AUR	61
Gambar 4. 12 Proses Pembuatan Ballast.....	61
Gambar 4. 13 Hasil Pembuatan Ballast.....	62
Gambar 4. 14 Skematik Rangkaian Elektronika Sistem <i>Mainboard</i>	63
Gambar 4. 15 Hasil <i>Routing</i> PCB Sistem <i>Mainboard</i>	63
Gambar 4. 16 Hasil Cetak <i>Mainboard</i> Beserta Komponen Pendukung	64
Gambar 4. 17 <i>Wiring Electric</i> pada <i>Electric Tray</i>	64
Gambar 4. 18 <i>Wiring Electric</i> pada <i>Enclosure</i> Bawah	65
Gambar 4. 19 Perakitan Seluruh Sistem Perangkat Keras dan Penggerak AUR..	66
Gambar 4. 20 Perakitan Sistem Ballast Tetap pada <i>Prototype AUR</i>	66
Gambar 4. 21 Proses Perendaman <i>Prototype AUR</i>	66
Gambar 4. 22 Proses Pengambilan <i>Dataset</i>	67
Gambar 4. 23 Proses <i>Labelling Dataset</i>	70
Gambar 4. 24 Konfigurasi <i>Split Dataset</i>	70
Gambar 4. 25 <i>file data.yaml</i>	71
Gambar 4. 26 <i>Setting Jenis Runtime</i>	72
Gambar 4. 27 Menghungkan <i>File Google Drive</i>	73
Gambar 4. 28 Merubah Direktori Penyimpanan	73
Gambar 4. 29 <i>Install Pustaka Ultralytics</i>	73
Gambar 4. 30 Kode <i>Training YOLOv8</i>	73
Gambar 4. 31 Hasil <i>Training Model YOLOv8</i>	74
Gambar 4. 32 <i>Confusion Matrix</i> Hasil <i>Training Model YOLOv8n</i>	75
Gambar 4. 33 Intensitas Cahaya Pengujian <i>Real-time</i>	88
Gambar 4. 34 Proses Uji Deteksi Objek secara langsung.....	88

(Halaman ini sengaja dikosongkan)

DAFTAR NOTASI

x,y	: Koordinat pusat <i>bounding box</i> relatif terhadap <i>grid cell</i> .
w,h	: Lebar dan tinggi <i>bounding box</i> relatif terhadap ukuran gambar.
TP	: <i>True Positive</i>
TN	: <i>True Negative</i>
FP	: <i>False Positive</i>
FN	: <i>False Negative</i>
Precision	: Rasio TP terhadap jumlah prediksi positif (TP + FP)
Recall	: Rasio TP terhadap jumlah data positif sebenarnya (TP + FN)
F1-Score	: Harmonik rata-rata Precision dan Recall sebagai ukuran performa deteksi
Accuracy	: Proporsi total prediksi yang benar, baik positif maupun negatif
PWM	: <i>Pulse Width Modulation</i> , teknik kontrol kecepatan motor
ADC	: <i>Analog to Digital Converter</i> , konversi sinyal <i>analog</i> ke digital
GPIO	: <i>General Purpose Input/Output</i> pin pada <i>microcontroller</i>
SRAM	: <i>Static Random-Access Memory size</i> pada <i>microcontroller</i>
ROM	: <i>Read-Only Memory size</i> pada <i>microcontroller</i>
BLE	: <i>Bluetooth Low Energy</i> , fitur konektivitas di ESP32
ESP32	: <i>Microcontroller</i> ESP32 yang digunakan sebagai pusat kendali dan komunikasi
YOLOv8n	: Versi nano dari algoritma <i>object detection</i> YOLOv8
GPU	: <i>Graphics Processing Unit</i> , Pendukung utama algoritma YOLOv8

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Bersamaan berkembangnya zaman, pertumbuhan teknologi mengalami kemajuan terutama pada bidang robotika. Peristiwa tersebut terjadi pada era modern pada saat ini yang terpengaruh oleh kecerdasan buatan, tentunya pada robot bawah air. Robot bawah air memiliki kemampuan yang sangat bermanfaat bagi manusia terutama dalam kemampuan menyelamnya yang dapat mendukung manusia meneliti sumberdaya laut (Nurhadi, 2019).

Robot bawah air atau *Autonomous Underwater Robot* (AUR) atau bisa disebut juga *Autonomous Underwater Vehicle* (AUV) dapat diterapkan pada sektor air tergantung pada kebutuhan yang diinginkan. Untuk menunjukan perkembangan didunia robotika khususnya pada sektor bawah air, IEEE OES (*Ocean Engineering Society*) *Singapore* dengan dukungan *National University of Singapore* dan *Singapore Polytechnic* menyelenggarakan *Singapore Autonomous Underwater Vehicle Challenge* (SAUVC). Terdapat beberapa persamaan antara kompetisi SAUVC dengan kompetisi Kontes Robot Indonesia (KRI) pada divisi Kontes Robot Bawah Air Indonesia (KRBAI) yaitu pada lintasan KRBAI yang diadaptasi dari SAUVC dan mengharuskan AUR untuk bergerak tanpa adanya kendali manusia. Kegiatan kompetisi SAUVC merupakan sebuah ajang pembelajaran yang hebat bagi para peserta untuk merasakan tantangan rekayasa sistem AUV dan mengembangkan keterampilan di bidang terkait, yaitu teknik mesin, listrik, dan perangkat lunak..

Konsep dari SAUVC adalah para peserta ditantang untuk membuat *prototype Autonomous Underwater Robot* (AUR) dengan seluruh bangunan pada *prototype* dapat menyelam di air. Dari desain yang sudah dibentuk AUR harus dapat melayang, meluncur, dan mendorong serta manuver digerakan melalui propeller dengan kecepatan tertentu. AUR memiliki misi yang sesuai pada pedoman *rules* SAUVC 2025. Lintasan terbentuk dari beberapa objek seperti *gate* dengan warna

merah, hitam , dan hijau (dalam satu objek), *flare* dengan warna biru, merah, kuning, dan orange (dalam empat objek yang berbeda), *drum* dengan warna merah.

Pada *Singapore Autonomous Underwater Vehicle Challenge* (SAUVC) terdapat waktu persiapan 10 menit, 15 menit untuk untuk menjalankan 4 misi yang harus diselesaikan, Pertama *Navigation* yaitu AUR memperlihatkan kemampuan untuk menyelam dari *starting zone* dan bergerak menghindari *flare* orange hingga melalui *gate*, kedua *Target Acquisition* yaitu AUR memperlihatkan kemampuan untuk mendeteksi dan memperoleh target diantara beberapa *drum* kemudian menjatuhkan bola pada target *drum* yang telah dipilih, Ketiga *Target Reacquisition* yaitu memperoleh kembali bola yang telah dijatukan pada *drum* yang telah dipilih dan harus dibawa hingga akhir pertandingan, Keempat *Communication & Localiation* yaitu memperlihatkan kemampuan untuk mendeteksi ketiga *flare* (*flare* merah, *flare* biru, *flare* kuning) dan menabrak *flare* yang terdapat bola *golf* pada atas *flare*. Setelah menjalankan misi pertama, AUR diperbolehkan menjalankan misi kedua hingga keempat secara tidak urut, AUR dianggap menyelesaikan misi jika sudah kembali kepermukaan.

AUR yang dikompetisikan dalam SAUVC dapat dikendalikan secara otomatis dan manual. Kendali Manual dapat dilakukan dengan mengkoneksikan AUR dengan menggunakan *Tether ROV Cable* melalui *Aviation Plug* pada bagian belakang AUR dan *Tether ROV Cable* dihubungkan *Tether Interface Board*, selanjutnya *Tether Interface Board* akan dikoneksikan ke Laptop menggunakan kabel LAN dan laptop perlu dikoneksikan ke *Joy Stick*, Selebihnya AUR dapat melakukan pengambilan data dan uji coba manuver dengan kendali manual. Ketika melakukan kendali Otomatis maka penunjang kendali manual harus didilepaskan karena kendali otomatis hanya ditujukan untuk menjalankan misi dari start hingga finish. Kendali otomatis merupakan syarat utama dalam melakukan kompetisi. Dari permasalahan tahun kemarin, faktor yang menyebabkan kegagalan finish adalah keterlambatan deteksi target menggunakan sensor kamera. Faktor tersebut banyak terjadi karena lintasan yang terletak diluar ruangan yang menyebabkan intensitas cahaya berubah-ubah, kegagalan tersebut terjadi berkali-kali sehingga menyebabkan AUR tidak bisa mencapai finish bahkan kegagalan terjadi dipertengahan waktu kompetisi.

Faktor Intensitas cahaya sangat mempengaruhi deteksi objek, dari analisa perubahan intensitas cahaya berpengaruh pada model warna *Hue, Saturation, Value* (HSV) dalam pengolahan citra. Ruang warna HSV memberikan hasil yang lebih baik dibandingkan dengan RGB karena lebih toleran terhadap perubahan cahaya. Hal ini penting untuk memastikan keakuratan dalam mendeteksi objek dalam kondisi pencahayaan yang berbeda. (Oni dkk., 2021)

Berdasarkan uraian di atas bahwa ketepatan pendeteksian warna objek pada lintasan sangat berpengaruh pada arah gerak AUR. Oleh karena itu, Tugas Akhir ini berfokus pada pembuatan sistem yang mampu mendeteksi warna lintasan secara akurat agar AUR dapat bergerak dengan baik dan mencapai garis finish tanpa menabrak lintasan satupun. Metode yang digunakan dalam sistem ini adalah *You Only Look Once* (YOLO). Ini adalah salah satu pendekatan deteksi objek berbasis *Deep Neural Network* (DNN) yang paling matang dengan kinerja yang sangat baik dalam hal kecepatan dan akurasi (Fang dkk., 2020) Selain itu, algoritma YOLO juga dapat digunakan untuk mendeteksi sebuah objek secara real-time. Dengan menggunakan algoritma ini, maka tidak diperlukan kalibrasi ulang setiap kali program akan dijalankan, sehingga dapat menghemat waktu saat perlombaan berlangsung. Hal ini karena sistem deteksi YOLO berbeda dengan algoritma yang lain, dimana *dataset* dari benda yang akan dideteksi dikumpulkan terlebih dahulu kemudian dilakukan labelling untuk mengklasifikasikan objek. Lalu dilakukan training dataset untuk menghasilkan *file weights*, yang mana *file* tersebut akan digunakan dalam algoritma pemrograman. Sehingga dengan penerapan metode YOLO pada algoritma pemrograman kamera dapat mendeteksi warna objek pada lintasan dengan baik dan melewati lintasan tanpa menyentuh pembatas sampai garis finish.

1.2 Rumusan Masalah

Dengan adanya latar belakang tersebut, maka dapat dirumuskan permasalahan yang akan dibahas pada Tugas Akhir ini yaitu :

1. Bagaimana *Autonomous Underwater Robot* (AUR) bisa mendeteksi lintasan sesuai dengan aturan lomba pada SAUVC 2025?

2. Bagaimana tingkat akurasi keseimbangan pada *Autonomous Underwater Robot* (AUR)?
3. Bagaimana tingkat akurasi metode *You Only Look Once* (YOLOv8) dalam mendeteksi citra lintasan SAUVC 2025 secara real-time dengan *webcam*?

1.3 Tujuan

Berdasarkan rumusan masalah tersebut, maka penelitian ini memiliki beberapa poin tujuan diantaranya adalah :

1. *Autonomous Underwater Robot* (AUR) dapat mendeteksi bentuk objek menggunakan metode *You Only Look Once* (YOLOv8) sesuai dengan aturan lomba pada SAUVC 2025.
2. *Autonomous Underwater Robot* (AUR) dapat berjalan secara seimbang pada lintasan.
3. Mengetahui tingkat akurasi sistem identifikasi warna objek pada lintasan *Autonomous Underwater Robot* (AUR) menggunakan metode *You Only Look Once* (YOLOv8) dari hasil citra yang ditangkap oleh *webcamera*.

1.4 Manfaat

Dengan adanya penelitian Tugas Akhir ini diharapkan dapat memberikan manfaat diantaranya adalah :

1. Metode *You Only Look Once* (YOLOv8) dapat diterapkan pada *Autonomous Underwater Robot* (AUR) untuk mendeteksi seluruh citra lintasan SAUVC 2025.
2. Membuktikan bahwa keseimbangan *Autonomous Underwater Robot* (AUR) memiliki akurasi tinggi.
3. *Autonomous Underwater Robot* (AUR) dapat membuktikan bahwa intensitas cahaya yang rendah dapat mendeteksi warna pada lintasan SAUVC 2025.
4. Dapat digunakan sebagai referensi untuk penelitian selanjutnya.

1.5 Batasan Masalah

Agar analisis dan pembahasan pada Tugas Akhir ini dapat lebih terarah dan fokus, maka diberikan batasan masalah diantaranya adalah :

1. *Autonomous Underwater Robot* (AUR) hanya mendeteksi objek tertentu pada lintasan yang berwarna sesuai dengan aturan SAUVC 2025.
2. Semua bagian *Autonomous Underwater Robot* (AUR) harus bisa menyelam dibawah permukaan air.

(Halaman ini sengaja dikosongkan)

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Penelitian Terdahulu

Penelitian sebelumnya dengan judul “Identifikasi Warna Lintasan Buoy Pada *Autonomous Submarine Surface Vehicle* (ASSV) Di Event Kontes Kapal Cepat Tak Berawak Nasional (KKCTBN) Menggunakan Metode *You Only Look Once* (YOLO)” yang ditulis oleh Faiz Romadhoni menyebutkan bahwa untuk meningkatkan komputasi sehingga program dapat bekerja dengan cepat dan efisien diperlukan Mini PC dengan spesifikasi yang tinggi khususnya pada GPU serta memperhatikan intensitas cahaya yang menenerangi objek dan memperbanyak *dataset*. Pada penelitian yang dilakukan sebelumnya membahas tentang pergerakan manuver pada *Autonomous Submarine Surface Vehicle* (ASSV) menggunakan sensor kamera dan Aktuator Thruster, untuk mendeteksi objek menggunakan metode *You Only Look Once* (YOLO). Penelitian ini menggunakan YOLOv4 dilakukan diatas permukaan air dan telah dilakukan beberapa kali percobaan deteksi objek dari mulai intensitas cahaya yang tinggi hingga rendah dan didapatkan hasil deteksi yang baik yaitu 98% untuk deteksi *buoy* hijau dan 100% untuk *buoy* merah (Romadhoni dkk., t.t.).

Dari kajian penelitian sebelumnya terdapat poin yang sesuai dengan rencana Pengaplikasian metode YOLOv8 untuk deteksi objek lintasan *Autonomous Underwater Robot* (AUR) pada event SAUVC. Peneliti akan menggunakan metode YOLOv8 untuk mendeteksi citra lintasan menggunakan arena *Singapore Autonomous Underwater Vehicle Challenge* (SAUVC) dengan tujuan agar AUR dapat mendeteksi citra lintasan dari awal start hingga finish dengan intensitas cahaya yang minimum tanpa adanya objek yang tidak terdeteksi sehingga AUR mampu melewati lintasan dengan baik dan konsisten.

2.2 Pengolahan Citra

Pada umumnya pengolahan citra digital adalah proses pengolahan citra digital menggunakan komputer untuk meningkatkan kualitas, mengekstraksi

informasi, atau menghasilkan representasi baru dari citra yang diinginkan. Citra yang dimaksud gambar, foto, atau tampilan yang lain berupa tampilan dua dimensi yang menunjukkan suatu bentuk objek. Citra digital dapat berbentuk rentetan angka-angka secara dua dimensional. Adapun beberapa operasi pada pengolahan citra yang digunakan untuk:

1. Image enhancement yaitu memperbaiki kualitas visual agar mudah dianalisa oleh sistem. Contohnya adalah peningkatan kualitas citra dengan memperjelas detail, mengurangi noise dan penyesuaian kontras.
2. Image restoration yaitu pemulihan citra yang rusak agar mendekati kondisi aslinya. Contohnya adalah memperbaiki citra yang terdegradasi akibat gerakan kamera atau objek.
3. Image segmentation yaitu memisahkan citra menjadi beberapa bagian berdasarkan elemen tertentu. Contohnya adalah mengelompokkan citra menjadi beberapa kelompok agar mudah dianalisis .
4. Image compression yaitu mengurangi ukuran citra tanpa mengorbankan kualitas citra. Contohnya adalah mengubah citra menjadi format tertentu agar menghemat penyimpanan.
5. Image fusion yaitu menggabungkan dua sumber citra yang berbeda menjadi satu citra tunggal. Contohnya adalah menggabungkan citra dari inframerah dan cahaya untuk pengawasan malam
6. Image reconstruction yaitu membangun kembali citra mentah dari perolehannya yang tidak utuh. Contohnya adalah memulihkan data citra dari transmisi yang terganggu.

2.3 Pixel, Resolusi dan Intensitas

Pixel adalah suatu bentuk komponen terkecil berupa titik yang jika digabungkan dapat membentuk suatu tampilan gambar utuh. Setiap *pixel* memiliki kecerahan dan warna tertentu. jika terdapat banyaknya warna pada suatu citra dapat disebut dengan intensitas. Pada umumnya terdapat beberapa jenis yang intensitas yang digunakan pada pengolahan citra seperti *Grayscale* (gradasi abu-abu), 16 juta warna (*True color*), 256 warna (*High color*), Hitam dan putih (*Black and white*), dan Kedalaman warna (*Color depth*). Pada umumnya gambar akan semakin bagus

ketika terdapat banyak warna yang menyusunnya. Terdapat jumlah maksimum gambar yang bergantung pada format file yang digunakan seperti pada *Portable Network Graphics* (JPEG) menggunakan 16 juta warna sebagai penyusunnya dan *Graphics Interchange Format* (GIF) menggunakan 256 warna sebagai penyusunnya.

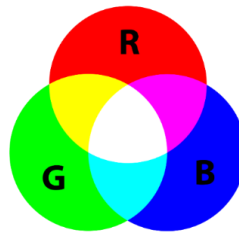
	0	1	2	3	.	.	n
0							
1							
2							
3							
.							
.							
m							

Gambar 2. 1 Koordinat *pixel*
(anjaruntoro.wordpress.com, 2025)

Pada Gambar 2.1 Koordinat pixel terdapat resolusi atau jumlah *pixel* perdaerah. Bentuk kualitas pada gambar dapat ditentukan oleh ukuran resolusi. Jika gambar diubah menjadi besar maka akan mengecilnya resolusi yang berakibat pada menurunnya kualitas gambar. Ketika gambar memiliki resolusi yang tinggi, maka gambar memiliki kemampuan diperbesar dan hasilnya tidak ada penurunan kualitas pada gambar.

2.4 Model Warna RGB

RGB merupakan suatu warna primer atau dasar yang terdiri atas 3 buah warna yaitu *red* (merah), *green* (hijau), dan *blue* (biru). Model warna RGB adalah model warna yang tersusun atas gabungan warna merah, hijau, biru yang menambah kuat pencahayaan primer. Jika dalam ruangan yang sama sekali tidak cahaya maka ruangan tersebut gelap total dan tidak ada cahaya yang bisa diserap oleh mata, maka bisa dikatakan RGB (0,0,0). Model RGB bekerja dengan menggabungkan intensitas tiga warna primer: merah, hijau, dan biru. Jika Anda meningkatkan intensitas biru ke nilai maksimumnya (255), objek lain dalam gambar akan muncul dalam nada biru yang lebih kuat, menggantikan properti warna aslinya (Rabbani dkk., 2021).

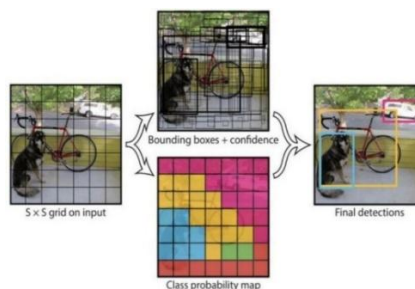


Gambar 2. 2 Model Gambar RGB
(Unitopledstrip, 2025)

Pada Umumnya model warna RGB banyak digunakan pada tampilan layar monitor, televisi, ponsel dan lain sebagainya. Semua perpaduan warna primer RGB dapat menghasilkan warna lain sebanyak 16 juta varian warna. Pada model warna RGB memiliki komposisi nilai yang berbeda, umumnya nilai warna tersebut bernilai 0 sampai 255 yang tergantung intensitas masing-masing warna. Bentuk pemodelan warna RGB dapat dilihat pada Gambar 2.2.

2.5 *You Only Look Once (YOLO)*

You Only Look Once (YOLO) merupakan revolusi dalam teknologi algoritma deteksi objek yang menggabungkan kecepatan dan akurasi tinggi dengan arsitektur jaringan saraf tunggal. Algoritma yang digunakan YOLO merupakan hasil pengembangan dari *Region-Based Convolutional Neural Network (R-CNN)*. Pengembangan algoritma sangatlah menguntungkan karena dapat meningkatkan frame persecond secara signifikan. Hal tersebut terjadi karena YOLO menggunakan satu langkah untuk melakukan *image Processing* dan R-CNN menggunakan dua tahapan untuk melakukan *image Processing*. *YOLO* memiliki waktu komputasi tercepat dibandingkan dengan *Faster R-CNN* dan *SSD*, tetapi akurasinya sedikit lebih rendah daripada *SSD* (Indah Kusumawati dkk., 2021).



Gambar 2. 3 Deteksi Objek YOLO
(www.researchgate.net, 2020)

You Only Look Once (YOLO) bekerja berdasarkan Gambar 2.3, Algoritma YOLO deteksi objek yang digunakan sangat cepat dan efisien dan berbeda dengan metode deteksi objek lain. Untuk proses cara kerjanya seperti:

1. Pembagian gambar menjadi *grid*

YOLO memulai dengan membagi gambar yang sudah di *Input* menjadi ukuran *grid*. Contohnya adalah ketika gambar yang di *Input* adalah 416 x 416 dan *grid* atau $S = 13$, maka Setiap sel *grid* akan memiliki ukuran 416/13 x 416/13 atau 32 x 32 Pixel.

2. Prediksi oleh setiap sel *grid*

Setiap sel *grid* akan memprediksi beberapa hal setiap objek terdeteksi seperti *Bounding box* yang didalam terdapat setiap sel *grid* memprediksi *bounding box* yang menggambarkan lokasi objek dan berfungsi memprediksi koordinat pusat terhadap posisi sel *grid*.

3. Prediksi kelas dan Lokasi

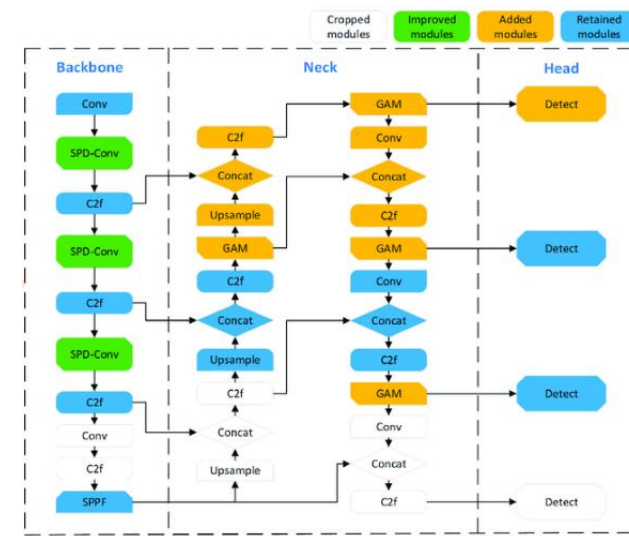
Pada Proses ini, Setiap sel dapat memprediksi lebih dari satu objek dengan menggunakan *bounding boxes* dengan memprediksi jumlah kotak yang berbeda dengan skor kepercayaan dan kelas objek yang terdeteksi.

4. *Non-Maximum Suppression* (NMS)

Penggunaan teknik *Non-Maximum Suppression* (NMS) berfungsi untuk menghapus prediksi yang kurang tepat dan proses ini terjadi setelah menghasilkan prediksi *bounding box* dan kelas objek untuk setiap *grid*. kemungkinan terdapat beberapa kotak yang tumpang tindih pada objek yang sama. NMS digunakan untuk mendapatkan tepi yang lebih akurat dengan menghapus piksel-piksel yang bukan puncak lokal sesuai arah gradien (Wicaksono dkk., 2024).

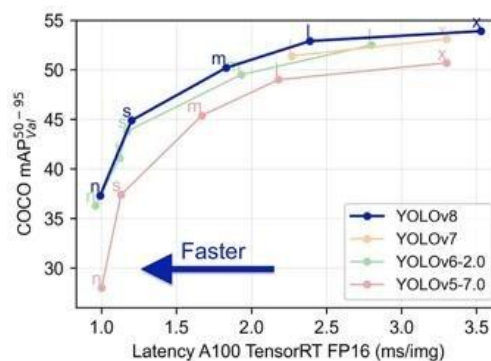
5. *Output* Deteksi

Proses ini merupakan akhir, Hasil yang didapatkan berupa *Output* seperti *Bounding box* berfungsi memisahkan lokasi objek dalam gambar, Kelas objek berguna untuk memisahkan jenis objek yang terdeteksi dan *Confident score* berfungsi menampilkan keyakinan bahwa objek tersebut benar terdeteksi.



Gambar 2. 4 Asitektur YOLOv8
(www.researchgate.net, 2024)

Pada Tugas Akhir ini menggunakan YOLO dengan versi YOLOv8, karena memiliki keunggulan sistem deteksi dan akurasi yang lebih baik dibanding versi sebelumnya. Kecepatan deteksi objek YOLOv8 dinilai sepuluh kali lebih cepat dibanding dengan YOLOv7 (Danyalson dkk., 2024).



Gambar 2. 5 Perbandingan *YOLOv8* dengan metode deteksi lainnya
(www.researchgate.net, 2023)

Pada Gambar 2.5 dapat diketahui bahwa YOLOv8 memungkinkan lebih baik dibanding dengan versi yang lain karena adanya peningkatan arsitektur, penggunaan dynamic anchor, augmentasi data yang lebih baik, serta optimasi pada perangkat keras modern seperti TensorRT. Hal ini memungkinkan YOLOv8 untuk mencapai latensi yang lebih rendah dan akurasinya lebih tinggi secara bersamaan.

2.6 Software Yang Digunakan

Dalam mendeteksi objek pada lintasan juga diperlukan software pendukung. Software yang digunakan memiliki beberapa fungsi seperti bahasa pemrograman, pendukung komputasi, library dan code editor. Berikut merupakan software pendukung yang digunakan

2.6.1 Python

Python merupakan bahasa pemrograman yang dirancang untuk memudahkan dalam pembacaan programnya dan efisiensi dalam pengembangan perangkat lunak. Python dikenal dalam sintaknya yang sederhana sehingga membuat cocok digunakan oleh pemula sampai profesional dalam bidangnya, termasuk dalam bidang pengolahan citra. Berikut merupakan bentuk logo dari yang dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Logo Python
(www.python.org, 2025)

Python didukung oleh komunitas yang besar dan bersifat open source serta dapat dijalankan di banyak platform sistem operasi sehingga memudahkan pengembang dalam membangun aplikasi lintas platform, memperoleh bantuan dari komunitas, serta memanfaatkan berbagai pustaka dan framework yang tersedia secara gratis. Python memiliki keunggulan sebagai bahasa pemrograman komputer yang sederhana dan fleksibel. Ini adalah perangkat lunak sumber terbuka (gratis) dan canggih, membuatnya ideal untuk pengembangan aplikasi di berbagai bidang (Fawwaz Al Maki & Korespondensi, 2022).

2.6.2 *Compute Unified Device Architecture (CUDA)*

Compute Unified Device Architecture (CUDA) adalah memainkan peran penting dalam mempercepat pengolahan citra dengan memanfaatkan kekuatan GPU. Dalam pengolahan citra, ada banyak operasi yang membutuhkan komputasi intensif, seperti filtering, transformasi, segmentasi, dan deteksi fitur. CUDA memungkinkan operasi-operasi ini dilakukan secara paralel, sehingga proses pengolahan citra bisa berjalan lebih optimal dibandingkan jika hanya menggunakan CPU. Penggunaan Nvidia Cuda dengan memanfaatkan arsitektur paralel pada GPU membuat pengolahan jadi lebih cepat dan efisien (Razak dkk., 2021). Bentuk Logo CUDA dapat dilihat pada Gambar 2.7.



Gambar 2. 7 Logo NVidia CUDA
(catalog.ngc.nvidia.com, 2023)

Pada umumnya, CUDA dalam pengolahan citra memiliki beberapa kegunaan seperti percepatan operasi dasar pada operasi citra, transformasi geometris, segmentasi citra, dan lain sebagainya. Dalam konteks YOLOv8, CUDA digunakan untuk mempercepat proses pelatihan dan inferensi model deteksi objek, memungkinkan pemrosesan data yang lebih cepat dan efisien

2.6.3 *Open Computer Vision (OpenCV)*

Open Computer Vision (OpenCV) adalah sebuah pustaka dengan sumber terbuka yang menyediakan berbagai fungsi untuk membaca, memproses dan menganalisa gambar serta video. Pustaka ini mendukung berbagai bahasa pemrograman seperti Python, C++, dan Java, serta dapat dijalankan pada berbagai platform sistem operasi seperti Windows, macOS, dan Linux. Bentuk logo dari OpenCV dapat dilihat pada gambar 2.8.

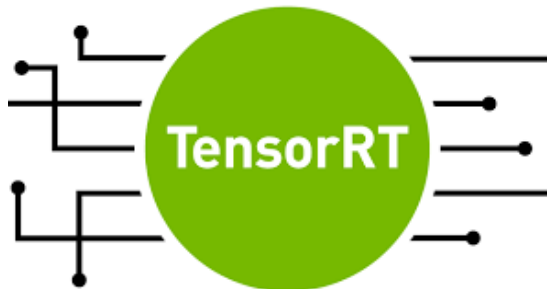


Gambar 2. 8 Logo OpenCV
(opencv.org, 2025)

Pada penggunaan pengolahan citra, OpenCV memiliki peran penting dalam YOLOv8 sebagai alat bantu utama untuk menangani *Input-Output* citra dan video, *pre-Processing*, serta menampilkan hasil deteksi. Kombinasi OpenCV dan YOLOv8 memungkinkan pengembangan aplikasi deteksi objek real-time yang cepat dan efisien. OpenCV memfasilitasi pemrosesan gambar, pengenalan pola, dan kalibrasi kamera, membuatnya sangat berguna untuk berbagai aplikasi dalam industri dan penelitian (Maulana dkk., 2024).

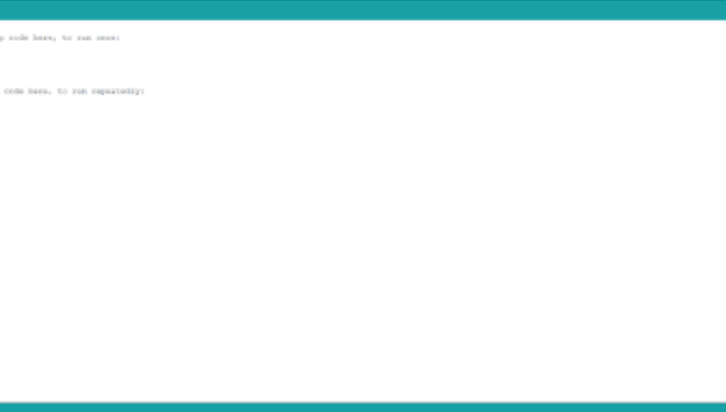
2.6.4 TensorRT

Tensor RT adalah sebuah pustaka yang digunakan untuk optimasi inferensi dari Nvidia yang dirancang untuk meningkatkan kecepatan inferensi model deep learning. Penggunaan TensorRT dalam proses inferensi YOLOv8 dapat meningkatkan kecepatan dan efisiensi kinerja model, sehingga menjadikannya lebih optimal untuk diterapkan pada aplikasi berbasis real-time (Muzammil & Indraswari, 2024). Berikut merupakan bentuk logo TensorRT yang bisa dilihat pada Gambar 2.9.



Gambar 2. 9 Logo TensorRT
(developer.nvidia.com, 2018)

2.6.5 Arduino IDE



00000_arduino: Arduino 1.8.19 (Windows Store 1.8.37.0)
File: Sketch Tools Menu

```

#include <Arduino.h>

void setup() {
  // put your setup code here, to run once:
}

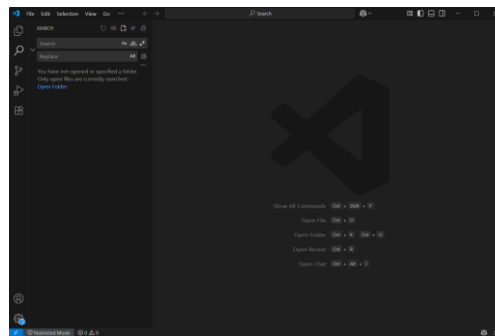
void loop() {
  // put your main code here, to run repeatedly:
}

```

Pada Arduino IDE struktur dasar program yang digunakan terdiri dari dua fungsi utama antara lain *Setup* yang digunakan untuk menjalankan sekali saat board dinyalakan serta pada *setup* ini bisa mengatur pin sebagai *Input* atau *Output* dan *Loop* digunakan menjalankan berulang-ulang selama board menyala serta pada *loop* dapat digunakan untuk menuliskan program yang ingin dijalankan secara terus menerus.

2.6.6 Visual Studio Code

Untuk melakukan *editing* perintah program Payton diperlukan sebuah editor program salah satunya Visual Studio Code (atau VS Code). VS Code adalah kode editor yang dapat diekstensi, dan perangkat lunak ini bersifat *open source*. VS Code memiliki jumlah kontributor tertinggi di GitHub, VS Code memungkinkan pengembang menambahkan bahasa baru ke lingkungan kerja, seperti *Python* dan lain-lain. Bentuk tampilan software VS Code dapat dilihat pada Gambar 2.11.



Gambar 2. 11 Tampilan *Software* VS Code
(Data Pribadi, 2025)

Visual Studio Code mudah digunakan dan dinilai *efficiency*(Hidayah & Rofiqoh, 2024).Kelebihan lainnya adalah arsitektur berbasis ekstensi, yang memungkinkan pengembang memperluas IDE sesuai kebutuhan mereka. Keunggulan utamanya terletak pada ringan dan fleksibelnya, memungkinkan penambahan komponen berurutan sesuai kebutuhan, menjadikannya pilihan yang populer di kalangan pengembang perangkat lunak.

2.7 Hardware Yang Digunakan

Dalam melakukan tugas akhir ini terdapat beberapa hardware yang digunakan meliputi sensor, aktuator, dan microcontroller. Adapaun penkjaslannya sebagai berikut.

2.7.1 Webcam

Webcam adalah kamera digital yang dihubungkan ke laptop atau komputer (Dwi Putra dkk., 2022). Fungsinya sebagai penangkap suatu citra yang kemudian di dikonversi sebuah komputer. *Webcam* banyak digunakan dalam bermacam-macam bidang seperti industri, pendidikan, dan keamanan. Umumnya webcam

digunakan pada sistem deteksi gerak untuk memberikan peringatan pekerja, kelas daring untuk pembelajaran online, deteksi identitas untuk memverifikasi identitas pengguna dan lain sebagainya.



Gambar 2. 12 *Webcam* Logitech C930E
(Logitech, 2025)

Pada Tugas Akhir ini *webcam* adalah peralatan yang paling dibutuhkan karena digunakan untuk sensor yang menangkap citra lintasan SAUVC agar AUR dapat berjalan dari start hingga finish dengan lancar. *Webcam* yang digunakan pada Tugas Akhir saat ini adalah Webcam Logitech C930E dengan spesifikasi bidang pandang diagonal sebesar 90° kamera 3 mega pixel dan resolusi 1920 x 1080 piksel.

2.7.2 *Pressure Sensor*

Pressure sensor adalah suatu alat yang digunakan untuk mengukur kedalaman suatu perairan menggunakan sistem tekanan yang akan diolah datanya menggunakan *microcontroller*. *Pressure* sensor bisa digunakan dibidang industri energi, Pengelolaan sumber daya air, militer. adapapun beberapa penggunaan *Depth* sensor seperti pengukuran kedalaman pada bendungan, pengeboran minyak dan gas untuk menentukan posisi pada bawah laut, navigasi kapal selam untuk menentukan posisi optimal kapal selam, dan lain sebagainya.



Gambar 2. 13 *Pressure Sensor* MS5837-30BA
(Rovmaker, 2025)

Pada Tugas Akhir ini *Pressure Sensor* digunakan sebagai pengukur kedalaman yang berfungsi untuk mempertahankan kedalaman AUR ketika dibawah permukaan air. *Pressure Sensor* yang digunakan pada Tugas Akhir ini adalah *Pressure Sensor MS5837-30BA* yang memiliki ketahanan tekanan maksimum 30 *bar* dengan menggunakan protokol komunikasi I2C serta tingkat ketelitian 0.2 mbar sehingga kedalaman yang dipertahankan lebih akurat.

2.7.3 *Attitude Angle Sensor*

Attitude Angle Sensor adalah suatu alat yang digunakan sebagai pengukur orientasi dan gerakan suatu objek dalam ruang tiga dimensi dan datanya akan diolah oleh mikrokontroller. *Attitude Angle Sensor* banyak digunakan dibidang sistem navigasi kapal, sistem militer, virtual dan augment reality seperti penggunaan *Attitude Angle Sensor* saat mengukur sudut orientasi untuk sistem navigasi kapal, kendali senjata yang presisi untuk sistem militer, mengukur gerakan perangkat untuk virtual dan augment reality, dan lain sebagainya.



Gambar 2. 14 HWT905 *Attitude Angle Sensor*
(Witmotion, 2025)

Pada Tugas Akhir ini *Attitude Angle Sensor* digunakan sebagai mengukur orientasi dan gerakan yang berfungsi mempertahankan arah gerakan AUR ketika bergerak didalam air. *Attitude Angle Sensor* yang digunakan pada Tugas Akhir ini adalah HWT905 *Attitude Angle Sensor* yang memiliki fitur mengukur *roll*, *pitch*, dan *yaw* dengan protokol komunikasi UART, RS232, dan I2C serta tingkat ketelitian 0.05°, sehingga memungkinkan mempertahankan arah gerakan yang akurat

2.7.4 Buck Converter

Buck converter adalah suatu pegangan yang menyesuaikan siklus kerja sakelar untuk menghasilkan tegangan keluaran yang lebih rendah daripada tegangan masukan (Rusdi, 2020). Cara kerja dari *Buck converter* dengan prinsip switching pada mosfet yang bekerja secara terus-menerus. Jika mosfet aktif, maka akan menyuplai beban dan menyimpan tegangan energi pada induktor dan jika mosfet mati, maka induktor melepaskan energi melalui dioda ke beban dan menjaga arus tetap mengalir. *Kontroller pulse with modulation (PWM)* mengatur duty cycle untuk menjaga tegangan *Output* tetap stabil sesuai pengaturan potensiometer dan kapasitor menghaluskan dan mengurangi ripple pada tegangan *Output*.



Gambar 2. 15 Modul XL4015 (*DC to DC Buck Converter*)
(Data Pribadi, 2025)

Pada Tugas Akhir ini *Buck converter* yang digunakan sebagai penurun *Output* tegangan dari baterai yang berfungsi untuk menyuplai daya *Push Pull Selenoid*. *Buck converter* yang digunakan adalah Modul XL4015 yang memiliki maksimum *Output* arus 5 Ampere dan tingkat efisiensi hingga 96%, sehingga meminimalkan kerugian daya ketika distribusikan.



Gambar 2. 16 Modul MP1584 (*DC to DC Buck Converter*)
(Data Pribadi, 2025)

Selain menggunakan Modul XL4015 yang bisa dilihat pada Gambar 2.14, Pada Tugas Akhir ini juga menggunakan *buck converter* modul MP1584 digunakan untuk menyuplai daya *microcontroller*. Gambar buck converter modul MP1584 bisa dilihat pada Gambar 2.15.

2.7.5 Mini PC

Mini PC (Mini *Personal Computer*) adalah komputer yang memiliki bentuk ukuran kecil dengan kemampuan setara dengan komputer dekstop pada umumnya. Perbedaan antara mini PC dengan komputer Destop adalah pada ukuran yang signifikan dan beratnya berbeda serta konsumsi daya yang diperlukan juga berbeda. Mini PC umumnya digunakan untuk pekerjaan ringan hingga menengah dengan konsumsi daya antara 10-40 watt. Adapun Spesifikasi Mini PC yang digunakan dilihat pada Tabel 2.1 dan Gambar 2.16

Tabel 2. 1 Spesifikasi Mini PC Jetson Orin NX

Parameter	Value
CPU	8-core Arm® Cortex®-A78AE v8.2 64-bit CPU 2MB L2 + 4MB L3, up to 2 GHz
Memory size	16 GB
Memory Types	16GB 128-bit LPDDR5
Memory Speed	102.4GB/s
Storage	Support external SSD NVMe M.2
Graphics Processing Unit	1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores
DC Input Voltage Supported	5-20 VDC
USB Port	3x USB 3.2 Gen2 (10 Gbps) 3x USB 2.0
Wi-Fi	Support External M.2 Key E
System	Linux
Dimension	92,6 x 99,8 x 28 (mm) (LxWxH)
Weight	Net 300gr



Gambar 2. 17 Mini PC NVidia Jetson Orin NX
(Nvidia, 2025)

Pada Tugas Akhir ini Mini PC yang digunakan sebagai pengganti komputer dekstop untuk memproses citra yang ditangkap oleh kamera. Hasil proses pengolahan data dari mini PC akan dikirimkan ke *microcontroller* menggunakan komunikasi serial. Untuk mendukung Tugas Akhir ini maka digunakan Mini PC NVidia Jetson Orin NX.

2.7.6 ESP32

ESP32 adalah sebuah microcontroller yang dirancang oleh *Espressif Systems* serta dilengkapi modul *wi-fi* dan *bluetooth* dengan kemampuan nirkabel dan performa tinggi. ESP32 terbentuk atas beberapa komponen seperti CPU, memori dan I/O digunakan menerima *Input* sensor, mengolah data dan menghasilkan *Output* yang diinginkan. Spesifikasi ESP32 dapat dilihat pada Tabel 2.2 dan Gambar 2.17

Tabel 2.2 Spesifikasi ESP32

Parameter	Value
<i>Processor</i>	<i>Dual-core Xtensa® 32-bit LX6, hingga 240 MHz, hingga 600 DMIPS</i>
<i>Memory</i>	520 KB SRAM, 448 KB ROM, 16 KB RTC SRAM
<i>Flash</i>	Hingga 16 MB (default 4 MB)
<i>Wi-Fi</i>	802.11 b/g/n (2.4 GHz), <i>mode</i> STA, AP, STA+AP

Parameter	Value
<i>Bluetooth</i>	<i>Bluetooth 4.2 BR/EDR dan BLE</i>
<i>Input/Output Pins</i>	34
<i>Analog to Digital Converter pins</i>	18
<i>Digital to Analog Converter pins</i>	GPIO25, GPIO26
<i>Pulse Width Modulation pins</i>	16
<i>Serial Peripheral Interface pins</i>	GPIO18 (SCK), GPIO19 (MISO), GPIO23 (MOSI), <i>Use any GPIO for Chip Select (CS)</i>
<i>Input/Output Voltage</i>	3.3V
<i>Input Voltage (nominal)</i>	4,8 - 5,5V
<i>DC Current per Input/Output pin</i>	12 mA
<i>Weight</i>	3 gr
<i>Dimension</i>	25,5x18x3 mm (LxWxH)



Gambar 2. 18 *Microcontroller ESP32 Wroom 32U*
(Data Pribadi, 2025)

Pada Tugas Akhir ini ESP32 berfungsi untuk penerima data dari beberapa sensor dan mengolah data yang diterima yang selanjutnya data tersebut digunakan untuk mengontrol aktuator pada AUR, seperti thruster dan solenoid. Untuk

mendukung Tugas Akhir ini maka digunakan *Microcontroller* ESP32 Wroom 32U yang bisa dilihat pada Gambar 2.17.

2.7.7 *Electronic Speed Controller (ESC)*

Electronic Speed Controller atau yang umumnya disebut ESC adalah bentuk suatu rangkaian elektronik yang digunakan mengontrol kecepatan suatu motor, arah putaran dan memberhentikan putaran pada *DC brushless* atau BLDC . Umumnya ESC digunakan pada drone sebagai pengontrol motor DC *Brushless*. ESC memiliki spesifikasi dengan menunjukkan kapasitas arusnya, seperti ESC 20A,30A,35A,45A, dan lain sebagainya.



Gambar 2. 19 ESC ROV MAKER
(rovmaker, 2025)

Pada Tugas Akhir ini menggunakan *Electronic Speed Controller* sebanyak 9 buah yang berfungsi untuk mengontrol motor T200 *Brushless* dan BLDC motor digunakan adalah ESC ROV MAKER dengan spesifikasi 30 ampere. ESC ROV MAKER memiliki keunggulan yaitu ukuran yang kecil, *support* dengan baterai 4S-6S, dan performa yang stabil, sehingga memungkinkan AUR untuk bergerak dengan performa stabil dan menjalankan serangkaian misi sesuai dengan aturan kompetisi.

2.7.8 *Motor Thruster*

Motor thruster adalah suatu aktuator yang dirancang untuk menghasilkan gaya dorong yang digunakan untuk manuver. Umumnya Motor Thruster memiliki

bagian utama seperti motor sebagai penggerak utama, propeller sebagai pendorong, ESC sebagai pengatur kecepatan, dan housing untuk kerangka. Pada penggunaan Motor Thruster sektor air dibutuhkan bahan bahan yang tahan air dan ringan.



Gambar 2. 20 T200 *Brushless ROV Thruster*
(rovmaker, 2025)

Pada Tugas Akhir ini Motor thruster yang digunakan produk T200 *Brushless ROV Thruster* dengan jumlah 8 buah yang diletakan pada bagian tepi AUR dan dipasang secara vertikal dan horizontal. Motor *thruster* ini digunakan pada AUR untuk melakukan manuver dengan menggunakan motor brushless DC yang dikontrol oleh ESC dengan menggunakan *Output* data yang diolah oleh *microcontroller*.

2.7.9 BLDC Motor

Brushless Direct Current atau bisa disebut BLDC motor adalah suatu aktuator yang menggunakan *Electronic Speed Controller* (ESC) untuk mengatur perputaran rotor melalui medan magnet yang dihasilkan oleh kumparan stator. BLDC Motor menghasilkan torsi yang baik dengan ukuran yang relatif kecil dan ringan.



Gambar 2. 21 BLDC Motor F2429
(Faradyi.com, 2025)

Pada Tugas Akhir ini BLDC Motor yang digunakan produk Farady F2429 yang dengan jumlah 1 buah yang diletakan pada bagian bawah AUR dan dipasang horizontal. BLDC Motor ini digunakan pada AUR untuk mekanisme *taker ball* atau pengambilan bola yang telah dijatuhkan pada *drum*. BLDC Motor ini memiliki maksimum arus sebesar 10.5 Ampere dan sudah *waterproof* sehingga aman digunakan untuk kondisi medan pada bawah air.

2.7.10 *Push Pull Solenoid*

Push pull solenoid adalah suatu aktuator yang menggunakan mekanisme elektromagnetik dua arah dengan mendorong menggunakan solenoid yang diberi tegangan dan menarik menggunakan spring ketika *solenoid* tidak ada tegangan. Umumnya *Push pull solenoid* digunakan pada kunci elektronik, mekanisme katup, mekanisme saklar, dan lain sebagainya.



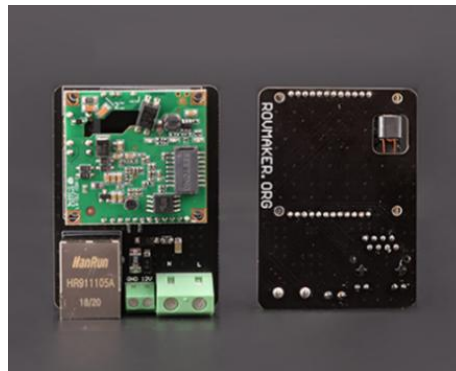
Gambar 2. 22 *Push Pull Solenoid* JF-0826B

(Data Pribadi, 2025)

Pada Tugas Akhir ini penggunaan *Push pull solenoid* digunakan untuk mekanisme *Drop Ball* atau pelepasan bola pada target yang telah dideteksi oleh webcam pada bagian bawah AUR. *Push pull solenoid* yang digunakan pada tugas akhir kali ini adalah Push Pull Solenoid JF-0826B yang memiliki tegangan kerja 12 VDC dan memiliki daya magnetik 20 Newton, sehingga pelepasan bola oleh *Push Pull Solenoid* akan berhasil.

2.7.11 Modul *Interface*

Modul *Interface* adalah sebuah rangkaian elektronik yang digunakan sebuah penghubung antara suatu perangkat dengan sistem utama. Umumnya Modul *Interface* berguna sebagai jembatan dengan menyederhanakan koneksi dan menghubungkan suatu komponen dengan alat yang berbeda. Penggunaan Modul *Interface* memudahkan pengguna dalam mengoperasikan suatu alat dengan jarak jauh.

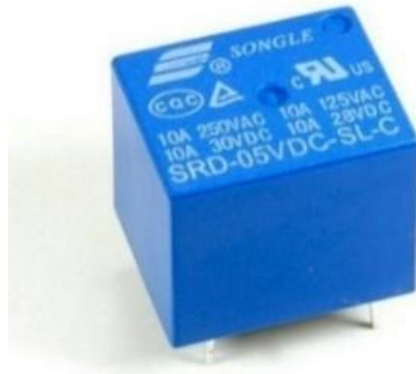


Gambar 2. 23 *Tether Interface Board*
(ROVMAKER, 2025)

Pada Tugas Akhir ini Modul *Interface* digunakan untuk mengoperasikan AUR secara manual untuk pengambilan data citra dan memantau manuver gerak AUR pada laptop. Pada Tugas Akhir *Interface Board* yang digunakan adalah 2 buah *Tether Interface Board* yang dapat dihubungkan pada Mini PC NVidia Jetson Orin NX dan pada Laptop. *Tether Interface Board* dapat menghubungkan AUR dengan Laptop lebih dari 500 meter menggunakan kabel khusus sehingga memudahkan operasi jarak jauh pada AUR.

2.7.12 Relay

Relay adalah suatu komponen elektronik yang menggunakan prinsip elektromekanis pada sistem kerjanya. Umumnya relay digunakan mengontrol listrik dengan arus yang melebihi kapasitas saklar dan menggunakan arus kecil sebagai pemicunya. Relay biasanya digunakan pada kendaraan sebagai kontrol kelistrikan, elektronik rumah tangga sebagai pengontrol pompa air, sistem keamanan sebagai kontrol alarm, dan lain sebagainya.



Gambar 2. 24 Relay Songle SRD-05VDC-SL-C
(Data Pribadi, 2025)

Pada Tugas Akhir ini Relay digunakan sebagai pengontrol *drop ball* pada bagian bawah AUR untuk menjatuhkan bola pada target yang ditentukan. Relay yang digunakan pada tugas Akhir ini adalah Relay Songle SRD-05VDC-SL-C yang memiliki konfigurasi *switch single pole double throw* dan maksimum arus 10 A, sehingga membuat kontrol *push pull Selenoid* akan berjalan secara normal.



Gambar 2. 25 Relay WM686 RL/180-12
(Data Pribadi, 2025)

Pada Tugas Akhir ini juga menggunakan 2 buah Relay WM686 RL/180-12. Relay pertama sebagai mekanisme kontrol terhadap seluruh sistem AUR yang mendapatkan distribusi listrik dari baterai dan relay kedua sebagai mekanisme kontrol terhadap Microcontroller, ESC, sensor dan aktuator. Bentuk Relay WM686 RL/180-12 dapat dilihat pada Gambar 2.25.

2.7.13 Switch

Switch atau biasa disebut saklar adalah perangkat mekanis dan elektronik yang digunakan untuk menghubungkan atau memutus aliran listrik pada sebuah rangkaian. Berdasarkan prinsip kerjanya switch dibagi menjadi dua yaitu *Normally Close* (NC) dan *Normally Open* (NO)(Pradiftha & Elektro, 2019).

1. *Normally Close* (NC), merupakan bentuk awal posisi switch dalam kondisi tertutup dan bisa mengalirkan listrik. ketika ditekan maka switch akan dalam kondisi terbuka dan menjadi memutuskan listrik.
2. *Normally Open* (NO), merupakan bentuk awal posisi switch dalam kondisi terbuka dan memutuskan listrik. ketika ditekan maka switch akan dalam kondisi tertutup dan menjadi mengalirkan listrik.



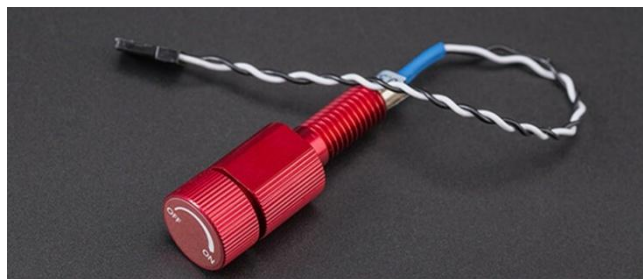
Gambar 2. 26 *Toggle Switch* MTS-102
(Data Pribadi, 2025)

Pada Tugas Akhir ini 2 buah Switch yang digunakan sebagai pengontrol yang fungsinya untuk memutus dan menghubungkan aliran listrik pada Relay 1 yang tugasnya mendistribusikan seluruh listrik ke semua sistem AUR dan Relay 2 sebagai pemutus dan penghubung rangkaian *Microcontroller*, ESC, sensor dan aktuator. Switch yang digunakan pada Tugas Akhir ini adalah *Toggle Switch* MTS-102 yang memiliki konfigurasi *Single Pole Double Throw* (SPDT) dan maksimum arus yang dapat dihantarkan 6A, sehingga memudahkan dalam mengontrol AUR.



Gambar 2. 27 *Magnetic Switch MC-38*
(Data Pribadi, 2025)

Pada Tugas Akhir ini, *Magnetic Switch MC-38* digunakan sebagai Trigger pada ESP32 untuk menjalankan program manuver otomatis AUR. Bentuk *Magnetic Switch MC-38* dapat dilihat pada Gambar 2.25.



Gambar 2. 28 *M10 Underwater Switch*
(rovmaker, 2025)

Pada Tugas Akhir ini, *M10 Underwater Switch* digunakan sebagai pemutus dan penghubung Tether Interface board pada AUR. Bentuk *M10 Underwater Switch* dapat dilihat pada Gambar 2.26

2.7.14 Baterai *Li-ion*

Baterai *Li-ion* atau bisa disebut dengan baterai *Lithium Ion* adalah salah jenis baterai yang digunakan untuk menyuplai daya pada rangkaian elektronika. Baterai *Li-ion* bekerja berdasarkan reaksi kimia reversibel yang melibatkan ion lithitium diantara katoda dan anoda. Jadi ketika baterai *li-ion* digunakan ion lithium akan bergerak dari anoda ke katoda dan akan terjadi sebaliknya jika dilakukan

pengisian ulang. Selain itu, baterai Li-On dinilai sebagai pilihan yang ramah lingkungan karena tidak mengandung bahan berbahaya (Rahardi, Adam and Raharjo, 2022).



Gambar 2. 29 Li-ion Battery SONY VTC6 18650
(Data Pribadi, 2025)

Pada Tugas Akhir ini baterai Li-ion digunakan adalah *Li-ion Battery* SONY VTC6 18650 yang bisa dilihat pada gambar 2.27. Baterai *Li-ion* ini berfungsi sebagai supply daya pada seluruh sistem pada AUR. Baterai *Li-ion* harus dilakukan kombinasi menggunakan rangkaian seri dan paralel untuk mencukupi kebutuhan tiap komponen pada AUR. Untuk kebutuhan supply baterai dibentuk menjadi 3 *cell* dan 4 *cell* yang dapat dilihat pada Tabel 2.3. Konfigurasi 1 *cell* bernilai 4.2 Volt jika baterai dalam kondisi penuh dan jika baterai kondisi habis atau lemah bernilai 3.7 Volt. Adapun perhitungan yang dapat dilihat pada persamaan 2.1, 2.2, dan 2.3.

Tabel 2. 2 Spesifikasi distribusi penggunaan Baterai Li-ion

Parameter	Value
Baterai <i>Li-ion</i> 3 <i>cell</i> (Tanpa Buck Converter)	<ul style="list-style-type: none"> - Mini PC NVidia Jetson Orin NX (12V) - <i>Push Pull Solenoid</i> (12V) - <i>Tether Interface Board</i> (12V)

Parameter	Value
Baterai <i>Li-ion</i> 3 cell (Menggunakan Buck Converter)	- ESP32 Wroom 32U (3,3V) - Pressure Sensor MS5837-30BA (3,3V) - HWT905 Attitude Angle Sensor (5V)
Battery <i>Li-ion</i> 4 cell	- T200 Brushless ROV Thruster (16,8V) - ESC ROV MAKER (16,8V)

Distribusi baterai *Li-ion* dibutuhkan untuk menyesuaikan kebutuhan tiap komponen. karena distribusi yang baik akan membuat distribusi beban seimbang dan menghindari tidak ratanya siklus pengosongan yang sesuai aturan akibat penggunaan yang tidak merata.

$$1 \text{ cell baterai } Li-ion = 4,2 \text{ Volt} \quad (2.1)$$

$$3 \text{ cell baterai } Li-ion = 4,2 \times 3 = 12,4 \text{ Volt} \quad (2.2)$$

$$4 \text{ cell baterai } Li-ion = 4,2 \times 4 = 16,8 \text{ Volt} \quad (2.3)$$

Konfigurasi perhitungan cell baterai *li-ion* pada persamaan 2.1, 2.2, dan 2.3 dapat mempengaruhi tegangan kerja, kapasitas daya, arus yang dibutuhkan, serta umur perangkat. Baterai *Li-ion* yang dirancang dengan spesifikasi yang tepat akan memastikan kinerja yang optimal, keamanan, serta efisiensi energi dari perangkat

2.8 Komunikasi Serial

Komunikasi serial adalah metode komunikasi data di mana informasi dikirim sedikit demi sedikit secara berurutan melalui saluran komunikasi tunggal, seperti kabel atau jalur komunikasi lainnya. Tidak seperti komunikasi paralel, yang mengirimkan beberapa bit data secara bersamaan melalui beberapa jalur, komunikasi serial hanya menggunakan satu jalur data, membuatnya lebih sederhana dan lebih cocok untuk jarak jauh. Dalam komunikasi serial, data dipecah menjadi bit (0 dan 1) dan ditransmisikan secara individual pada kecepatan tertentu yang disebut baud rate. Agar komunikasi berfungsi dengan baik, pemancar dan penerima harus menggunakan baud rate yang sama. Jenis komunikasi serial meliputi:

2.8.1 Komunikasi *Serial Sinkron*

Komunikasi *serial sinkron* adalah metode transmisi data di mana transmitter atau pengirim dan receiver atau penerima menggunakan sinyal *clock* yang sama untuk menyinkronkan transmisi data. Dalam komunikasi ini, data dikirim sedikit demi sedikit secara berurutan dan pengirim mengirimkan sinyal *clock* bersama dengan data sehingga penerima tahu kapan harus membaca bit data berikutnya. Berikut ini adalah contoh komunikasi serial sinkron :

1. SPI (Antarmuka Periferal Serial) adalah metode komunikasi serial yang digunakan untuk mengirimkan data antara *microcontroller* dan perangkat seperti *register* geser, sensor, dan kartu SD. Karena menggunakan pertukaran data cepat, komunikasi SPI cocok digunakan untuk pengalokasian hardware dengan menggunakan sistem sinkronisasi didalamnya (Prakosa dkk., 2024).
2. I2C (*Inter-Integrated Circuit*) suatu protokol yang menggunakan sinyal *clock* untuk mengontrol komunikasi antara beberapa perangkat dalam sistem yang sama. Menggunakan I2C menghemat pin *microcontroller* dan memungkinkan desain perangkat keras yang lebih efisien (Dwinanto & Yulianto, 2024)

2.8.2 Komunikasi *Serial Asinkron*

Komunikasi *serial Asinkron* adalah Metode komunikasi di mana data dikirim secara berurutan (*serial*) tanpa menggunakan sinyal *clock* eksternal untuk menyinkronkan *transmitter* dan *receiver*. Berikut ini adalah contoh komunikasi *serial Asinkron* :

1. UART (*Universal Asynchronous Receiver Transmitter*) adalah salah satu protokol komunikasi serial asinkron yang paling banyak digunakan pada perangkat tertanam dan IoT. UART mengubah data paralel menjadi serial untuk transmisi melalui pin TX dan RX. Pada penggunaannya TX digunakan sebagai *transmitter* atau pengirim dan RX sebagai penerima atau Receiver serta *Clock* generator digunakan setting kecepatan transfer data (Yurkha Kusuma dkk., 2020).

2. *Port Serial* pada Komputer, adalah antarmuka yang digunakan untuk menghubungkan perangkat secara serial, memungkinkan pengiriman data satu per satu Umumnya banyak komputer menggunakan komunikasi serial asinkron melalui port serial (COM) untuk berkomunikasi dengan perangkat eksternal seperti modem dan printer.

BAB 3

METODOLOGI PENELITIAN

Pada bab ini akan membahas mengenai alur penelitian dari Tugas Akhir untuk menjawab rumusan masalah secara struktur. Pembahasan tersebut meliputi konsep penelitian, tahapan penelitian, perancangan desain, jadwal dan rencana anggaran penelitian. Berikut adalah penjelasan lebih rinci tentang metodologi penelitian pada Tugas Akhir dengan judul “Implementasi Metode YOLOv8 Untuk Deteksi Objek lintasan *Autonomous Underwater Robot* (AUR) Pada Event SAUVC”.

3.1 Konsep Penelitian

Pada bagian ini menjelaskan tentang konsep penelitian yang akan dilakukan. Konsep penelitian di gunakan sebagai landasan dalam merancang sistem mulai dari desain, perancangan, dan pengembangan sistem. Pada Tugas akhir ini penelitian berfokus pada pengimplementasian YOLOv8 pada *prototype Autonomous Underwater Robot* (AUR) sehingga dapat mendeteksi objek pada lintasan.

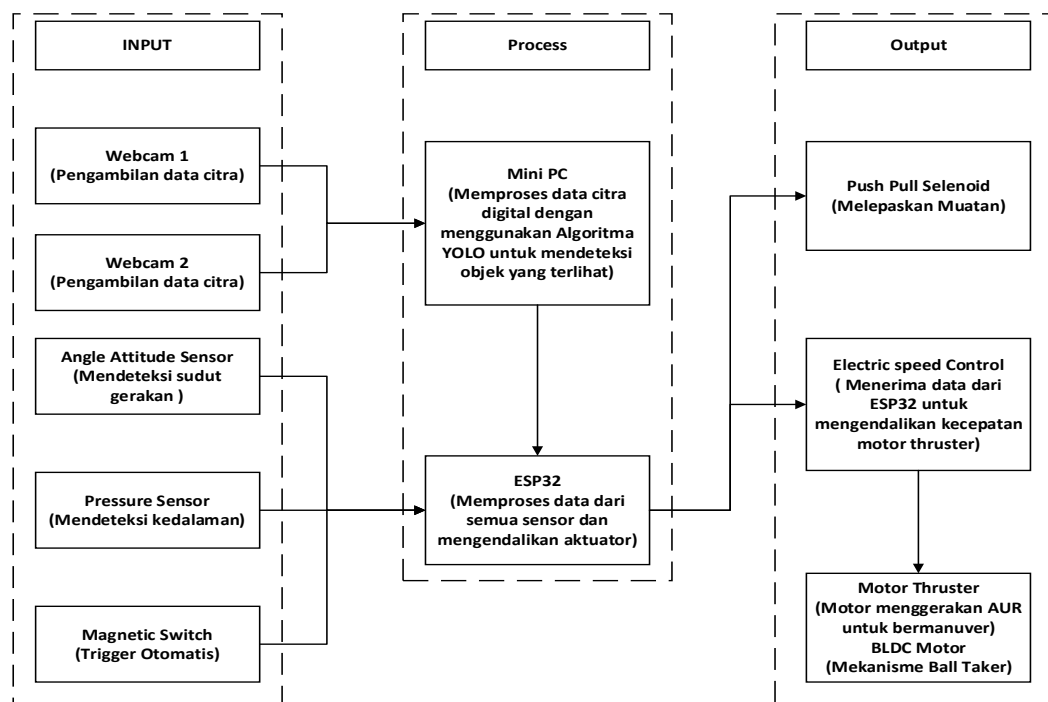
3.1.1 State Of The Art (Keterbaruan dari penelitian)

Tugas Akhir dengan judul “Implementasi Metode YOLOv8 Untuk Deteksi Objek lintasan *Autonomous Underwater Robot* (AUR) Pada Event SAUVC” merupakan sebuah penerapan yang dapat dilakukan pada *prototype Autonomous Underwater Robot* (AUR). Penelitian ini merupakan lanjutan dari penelitian sebelumnya dengan judul “Identifikasi Warna *Buoy* Menggunakan Metode *You Only Look Once Pada Unamed Surface Vehicle*” yang di lakukan oleh Faiz Romadhoni, dkk. Pada penelitian yang dilakukan oleh Faiz Romadhoni, dkk mendapatkan hasil efektif yaitu deteksi menggunakan algoritma YOLOv4 didapatkan akurasi 98% untuk *buoy* warna hijau dan 100% untuk *buoy* warna merah, hal tersebut yang melandasi penggunaan metode YOLO untuk deteksi penelitian selanjutnya. Untuk mendapatkan hasil yang lebih baik peneliti memilih YOLOv8 sebagai objek deteksi yang diharapkan memiliki akurasi yang baik

dengan kecepatan deteksi yang lebih cepat. Untuk keterbaruan peneliti mengimplementasikan metode deteksi YOLOv8 pada *prototype Autonomous Underwater Robot* (AUR) sesuai lintasan yang telah di tentukan dan menggunakan intensitas Cahaya yang minimum serta menambahkan sensor keseimbangan untuk menunjang pergerakan AUR dibawah air.

3.1.2 Prinsip Kerja Sistem

Prinsip kerja sistem pada penelitian ini merujuk pada aturan dasar bagaimana sistem dapat beroperasi. Pada umumnya sistem merupakan suatu kumpulan dari beberapa komponen yang dapat saling berinteraksi dengan target yang sama. Diagram blok sistem yang diaplikasikan pada Tugas Akhir ini dapat dilihat pada Gambar 3.1



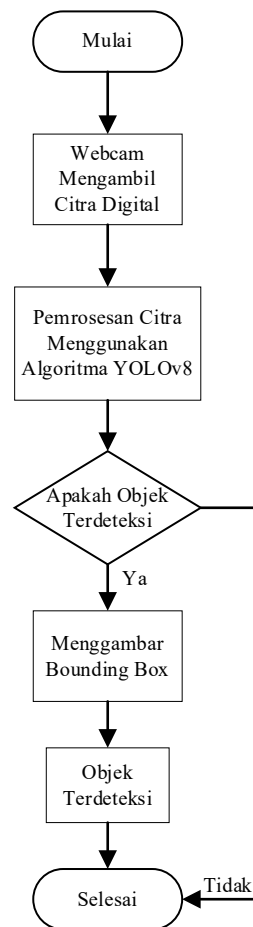
Gambar 3. 1 Diagram Blok Sistem

Dapat dilihat pada Gambar 3.1 diagram dibagi menjadi tiga bagian utama yaitu *Input*, *proses*, dan *Output*. *Input* terdiri dari *webcam 1*, *webcam 2*, *angle attitude sensor*, *pressure sensor*, dan *magnetic switch* yang digunakan sebagai pendeteksi citra lintasan, pengukur sudut kemiringan robot, pengukur kedalaman robot dan tombol menjalankan robot. Pada bagian proses terdapat *mini-PC*

digunakan untuk memproses citra dengan metode YOLOv8 *Object Detection* sebagai pendeteksi objek lintasan. Lalu hasil pembacaan objek yang terdeteksi akan dikirimkan ke *microcontroller*. *Microcontroller* berfungsi sebagai pengolah data dari beberapa sensor dan eksekutor untuk menggerakkan aktuator sesuai dengan data yang dikirim oleh beberapa sensor dan *mini-PC*. Bagian ketiga adalah *Output*, bagian ini terdapat tiga komponen yaitu *push pull selenoid* untuk membuka mekanisme *drop ball*, motor *thruster* sebagai penggerak utama *prototype* dan BLDC motor untuk mengambil bola yang telah dijatuhkan.

3.1.3 Sistem Deteksi Objek Dengan Metode YOLOv8

Pada Tugas Akhir ini, terdapat prinsip kerja dari pengolah citra dengan beberapa bentuk objek pada lintasan. Prinsip kerja sistem deteksi objek dapat digambarkan pada *Flowchart* 3.2



Gambar 3. 2 Diagram Alur Proses Deteksi Objek

Dari diagram Gambar 3.2 dapat diambil Kesimpulan, Webcam mengambil citra digital. Setelah citra digital diambil, akan diproses menggunakan algoritma YOLOv8. Dalam algoritma YOLOv8 terdapat beberapa proses antara konvolusi dan pooling. Proses konvolusi digunakan untuk mengekstrak fitur citra atau gambar sedangkan *pooling* digunakan untuk memperkecil dimensi citra tanpa menghilangkan fiturnya. Untuk memper jelas berikut adalah contoh proses konvolusi pada citra digital dengan ukuran 4x4 yang tersedia pada Gambar 3.3

1	0	0	1
0	1	1	0
0	1	1	0
1	0	0	1

Gambar 3. 3 *Matrix* Citra Digital 4x4

Citra di atas merupakan citra berukuran 4x4 sebagai citra masukan yang akan diproses konvolusi dengan menggunakan filter berukuran 3x3, yang akan menghasilkan citra berukuran 3x3. Untuk memperjelas proses konvolusi berikut adalah ilustrasi proses konvolusi tersedia pada Gambar 3.4

1	0	0	1
0	1	1	0
0	1	1	0
1	0	0	1

X

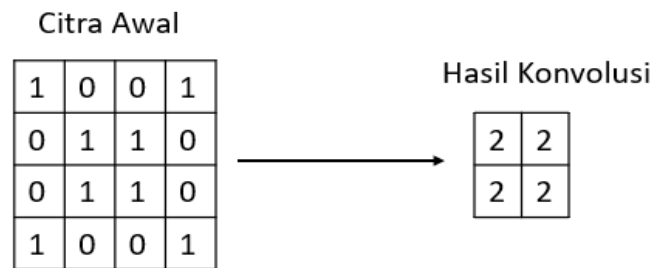
0	1	0
0	1	0
0	1	0

=

$$\begin{aligned}
 &(1 \times 0) + (0 \times 1) + (0 \times 0) + \\
 &(0 \times 0) + (1 \times 1) + (1 \times 0) + \\
 &(0 \times 0) + (1 \times 1) + (1 \times 0) + \\
 &= 2
 \end{aligned}$$

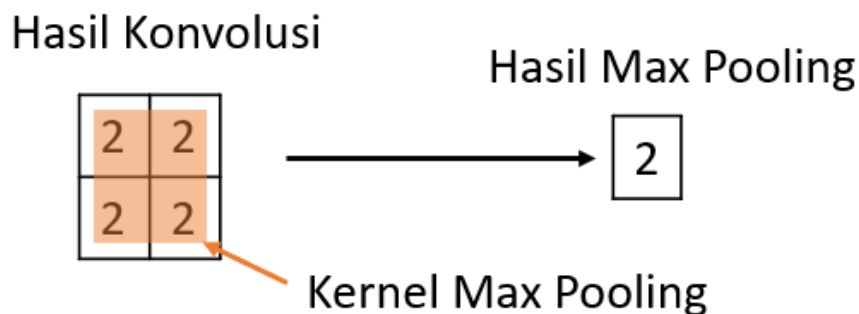
Gambar 3. 4 Ilustrasi Proses Konvolusi Citra

Pada tahap konvolusi seluruh citra akan di konvolusi dengan filter secara bertahap. Filter akan bergerak mengelilingi citra digital dan dilakukan proses konvolusi secara perlahan seperti pada Gambar 3.4. Dari proses konvolusi citra yang awalnya berukuran 4x4 akan berubah menjadi 2x2 tersedia pada Gambar 3.5



Gambar 3. 5 Hasil Proses Konvolusi Citra

Setelah citra melewati proses konvolusi maka Langkah selanjutnya citra dapat dilakukan *pooling*. *Pooling* yang paling sering digunakan biasanya ialah *maxpolling*. *Max pooling* merupakan operasi yang berfungsi untuk memper kecil ukuran citra dalam area *kernel* yang dipakai dengan cara mencari nilai maksimum. Berikut adalah proses *max pooling* pada citra 2x2 hasil dari proses konvolusi sebelumnya dengan ukuran *kernel max pooling* 2x2 tersedia pada Gambar 3.6

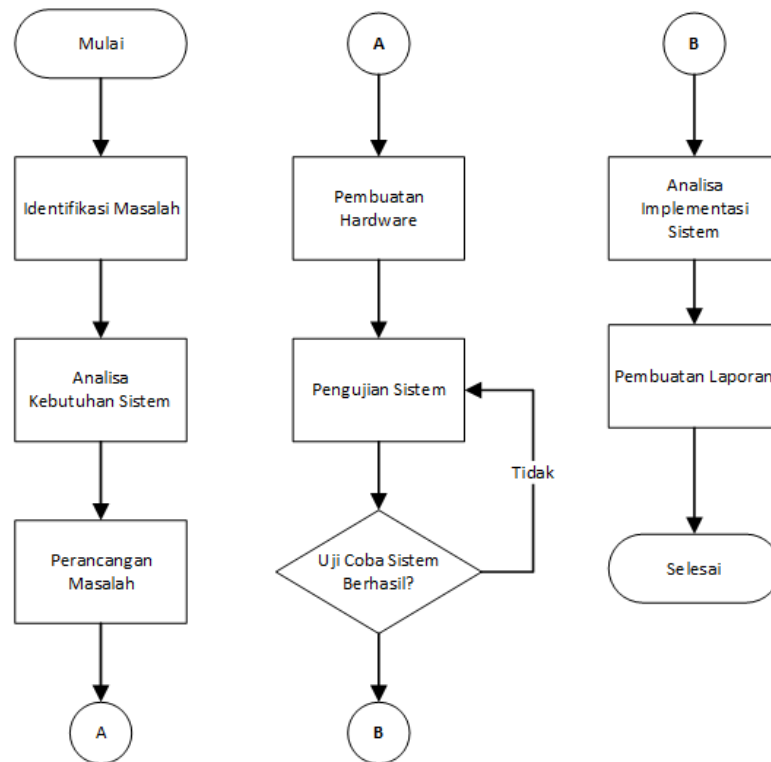


Gambar 3. 6 Hasil Proses Max Pooling Citra

Dari proses *max pooling* yang dilakukan pada citra awal dengan ukuran 2x2 diperoleh citra berukuran 1x1 dengan nilai citra 2 yang merupakan nilai tertinggi pada area *kernel max pooling* 2x2. Dengan demikian dari proses konvolusi dan *max pooling* yang merupakan dasar dari metode YOLOv8 dengan *Input* citra berukuran 4x4 maka diperoleh citra berukuran 1x1. Filter – filter dan operasi ini merupakan bagian dari pengolahan citra dengan menggunakan metode YOLOv8 yang tersimpan dalam bobot model. Bobot tersebut akan di gunakan sebagai data untuk memproses gambar sehingga objek pada gambar terdeteksi yang di tandai dengan adanya *bounding box* sebagai penanda, serta kelas dan kepercayaan objek yang terdeteksi.

3.2 Diagram Alur Penelitian

Pada bab ini akan dibahas tentang alur penelitian dari Tugas Akhir untuk melaksanakan proses dan menyelesaikan masalah secara terstruktur, serta analisa dan pembahasan. Bentuk pelaksanaan penelitian Tugas Akhir dapat dilihat pada alur 3.7



Gambar 3. 7 Alur Penelitian

3.3 Identifikasi Awal

Identifikasi awal perlu dilakukan sebelum melakukan penelitian. Langkah ini perlu dilakukan untuk menentukan suatu masalah saat yang akan dibahas saat melakukan dasar penelitian. Pada langkah ini juga bertujuan untuk menentukan target dan tujuan yang akan dicapai dalam sebuah penelitian. Penentuan langkah-langkah yang dilakukan pada tahap identifikasi awal adalah sebagai berikut.

3.3.1 Identifikasi Masalah

Dalam tahap ini peneliti mengidentifikasi permasalahan yang terjadi dalam *Singapore Autonomous Underwater Vehicle Challenge 2025 (SAUVC)*. Pada

SAUVC, robot dibuat oleh peserta kompetisi harus bisa kedap terhadap air dan bisa mendeteksi objek lintasan dibawah air. Pada kompetisi ini penggunaan *Webcam* dan *mini PC*, merupakan hal yang sangat penting karena kedua komponen yang mendukung agar AUR menyelesaikan misi dibawah air.

Berdasarkan permasalahan tersebut, Tugas Akhir ini mengambil judul “Implementasi Metode YOLOv8 Untuk Deteksi Objek lintasan *Autonomous Underwater Robot* (AUR) Pada Event SAUVC”. Harapannya metode ini dapat Diimplementasikan pada AUR agar dapat mendeteksi lintasan secara cepat dan tepat sehingga AUR dapat menyelesaikan semua misi.

3.3.2 Penetapan Tujuan dan Manfaat Penelitian

Pada yang telah dijabarkan di sub bab 3.3.1 berdasarkan identifikasi masalah, terdapat beberapa rumusann masalah yang terkait dengan penerapan metode YOLOv8 dan cara mendapatkan nilai akurasi deteksi objek yang optimal. Sehingga sistem ini bertujuan untuk mengimplementasikan YOLOv8 pada proses pengolahan citra yang diterapkan pada *Autonomous Underwater Robot* (AUR) untuk mendeteksi warna objek yang berbentuk lintasan dengan akurat. Agar dapat menyelesaikan semua misi dan mendapatkan predikat juara.

3.3.3 Studi Literatur

Studi literatur merupakan tahapan setelah menentukan rumusan masalah dan menentukan manfaat penelitian. Studi Literatur adalah proses proses mengumpulkan, meninjau dan menganalisis sebagai sumber informasi dari penelitian tertentu seperti jurnal, e-book, dan penelitian yang sebelumnya sudah dilakukan. Studi literatur bertujuan untuk mengetahui dasar teori, perkembangan, metode, latar belakang yang akan digunakan untuk menunjang dalam menyelesaikan tugas akhir ini.

3.3.4 Analisa Kebutuhan Sistem

Analisa dan kebutuhan sistem adalah suatu tahap yang bertujuan untuk mengetahui segala kebutuhan dari sistem yang akan digunakan dan memperhitungkan komponen yang akan digunakan agar sistem berjalan dengan baik. Adapun komponen yang digunakan pada sistem, sebagai berikut.

Tabel 3. 1 Jumlah Kebutuhan Sistem

No	Hardware	Jumlah
1	<i>Webcam Logitech C390e</i>	<i>2 Unit</i>
2	<i>Pressure Sensor MS5837-30BA</i>	<i>1 Unit</i>
3	<i>HWT905 Attitude Angle Sensor</i>	<i>1 Unit</i>
4	<i>Mini PC NVidia Jetson Orin NX</i>	<i>1 Unit</i>
5	<i>Microcontroller ESP32 Wroom 32U</i>	<i>1 Unit</i>
6	<i>Electric Speed Control</i>	<i>9 Unit</i>
7	<i>T200 Brushless ROV Thruster</i>	<i>8 Unit</i>
8	<i>Brushless DC motor</i>	<i>1 Unit</i>
9	<i>Push Pull Solenoid JF-0826B</i>	<i>1 Unit</i>
10	<i>Tether Interface Board</i>	<i>2 Unit</i>
11	<i>Toggle Switch MTS-102</i>	<i>3 Unit</i>
12	<i>Magnetic Switch MC-38</i>	<i>1 Unit</i>
13	<i>M10 Underwater Switch</i>	<i>1 Unit</i>

3.4 Desain dan Perencanaan Sistem

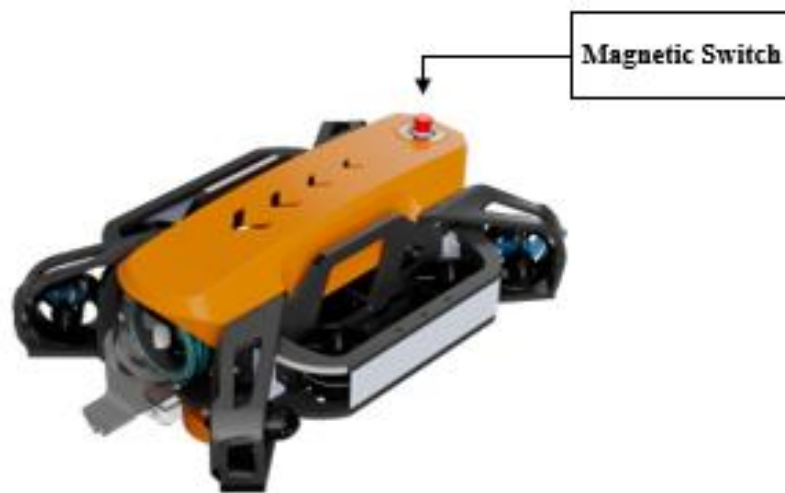
Desain dan perencanaan sistem adalah tahap yang bertujuan membuat gambaran visual agar diketahui bagaimana bentuk desain dan menetapkan kebutuhan agar tercukupinya komponen penunjang pada suatu sistem.

3.4.1 Desain Sistem Mekanik

Desain *prototype Autonomous Underwater Robot* (AUR) pada penelitian ini dibuat dengan panduan rules SAUVC 2025. Berikut merupakan ukuran dan dimensi AUR

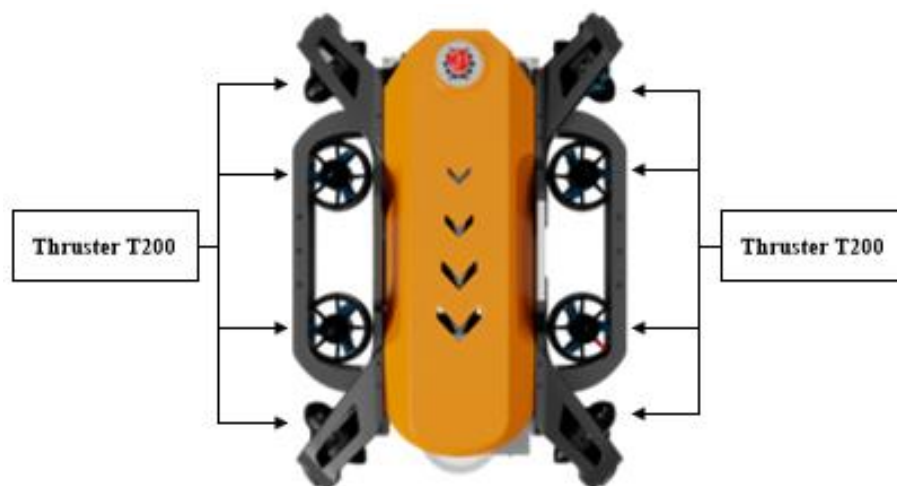
- Panjang : 64,7 cm
- Lebar : 44,4 cm
- Tinggi : 28,1 cm
- Berat : 19 Kg

Berikut merupakan desain *Autonomous Underwater Robot* (AUR) yang digunakan pada Tugas Akhir ini.



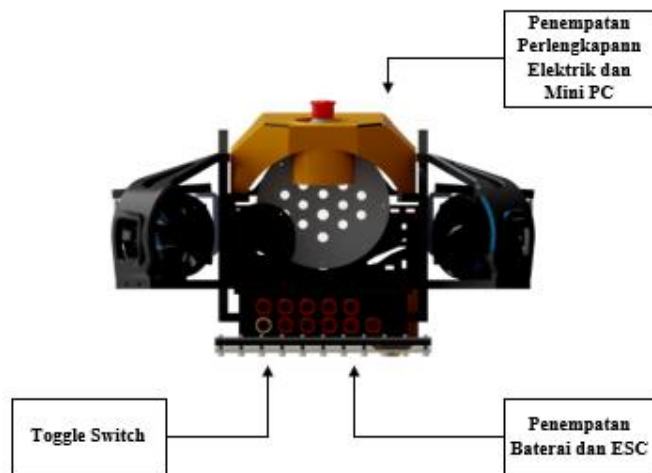
Gambar 3. 8 Desain AUR Tampak Isometri

Sesuai pada Gambar 3.8 tampak isometri, *magnetic switch* diletakkan pada bagian *ballast* bagian belakang dan pada bagian tengah *ballast* diberi lubang berbentuk segitiga berfungsi sebagai tempat keluarnya angin saat AUR menyelam ke bawah permukaan air.



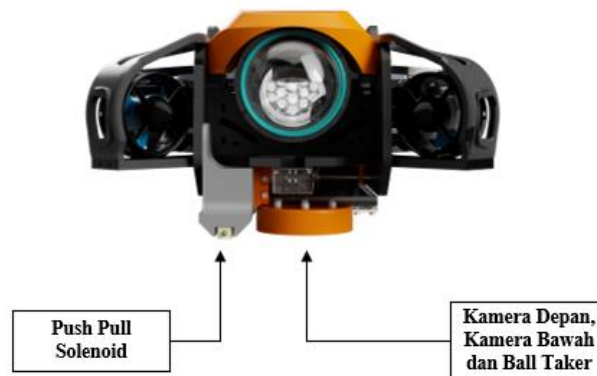
Gambar 3. 9 Desain AUR Tampak Atas

Sesuai pada Gambar 3.9 tampak atas, AUR memiliki delapan buah *thruster* yang digunakan untuk manuver. Terdapat 4 buah *Thruster T200* yang diletakkan horizontal yang berfungsi sebagai penunjang gerak AUR kearah maju dan mundur. Sedangkan 4 buah *Thruster T200* yang diletakkan vertical berfungsi sebagai penunjang gerak AUR kearah atas dan bawah.



Gambar 3. 10 Desain AUR Tampak Belakang

Sesuai pada Gambar 3.10 tampak belakang, terdapat komponen yang diletakkan dibagian *enclosure* utama seperti *microcontroller*, baterai *li-ion* 3 cell, HWT905, *mini* PC, dan komponen penunjang elektrik lainnya. Selain itu, terdapat komponen pada *enclosure* bawah seperti *Electric Speed Control* (ESC), baterai *li-ion* 4 cell dan relay. Untuk mengontrol semua sistem pada *enclosure* utama dan bawah diletakkan *toggle switch* pada bagian belakang AUR.



Gambar 3. 11 Desain AUR Tampak Depan

Sesuai pada Gambar 3.11 tampak depan, terdapat komponen yang diletakkan didalam pada bagian depan *enclosure* utama yaitu kamera depan (*webcam 1*). Pada bagian bawah terdapat juga kamera bawah (*webcam 2*), serta terdapat mekanisme *ball drop* menggunakan *push pull solenoid* dan mekanisme *ball taker* menggunakan BLDC motor.

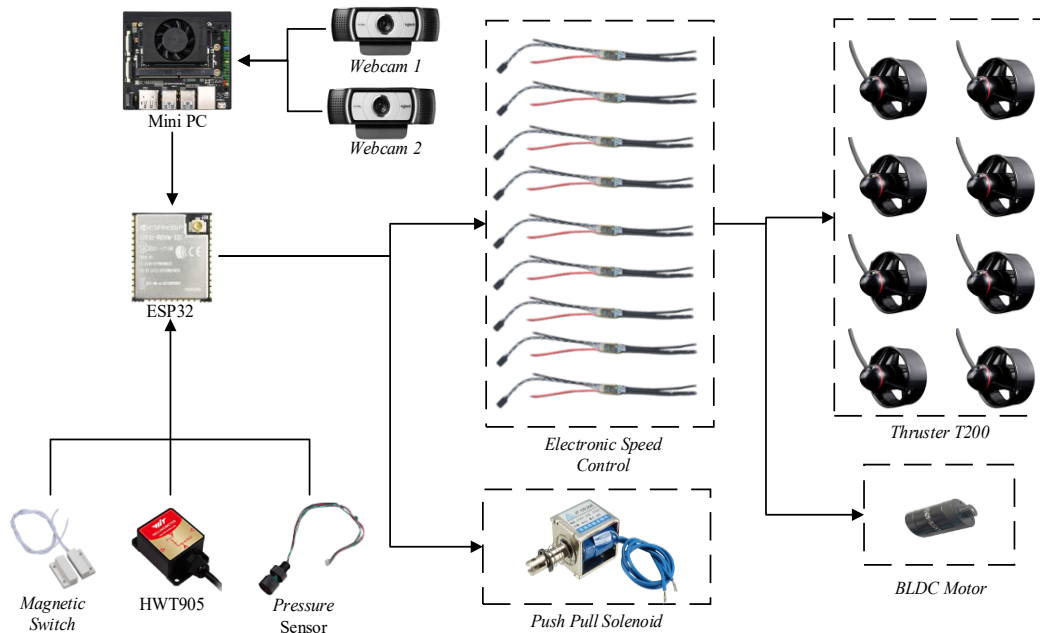


Gambar 3. 12 Desain AUR Tampak Samping

Sesuai pada Gambar 3.12 tampak samping, dapat dilihat bahwa AUR didesain dengan secara efisien dengan memberikan *handle* pegangan yang memudahkan AUR saat dibawa atau diangkat.

3.4.2 Perencanaan *Hardware*

Perancangan *hardware* bertujuan untuk menentukan bagaimana Hardware tersebut di susun dan di rangkai sehingga mampu berjalan sesuai dengan kebutuhan. Pada Tugas Akhir ini terdapat beberapa *hardware* yang terdiri atas beberapa komponen seperti sensor, aktuator, dan *microcontroller*. Berikut bentuk rancangan *hardware* yang digunakan dalam Tugas Akhir ini.



Gambar 3. 13 Perancangan Hardware

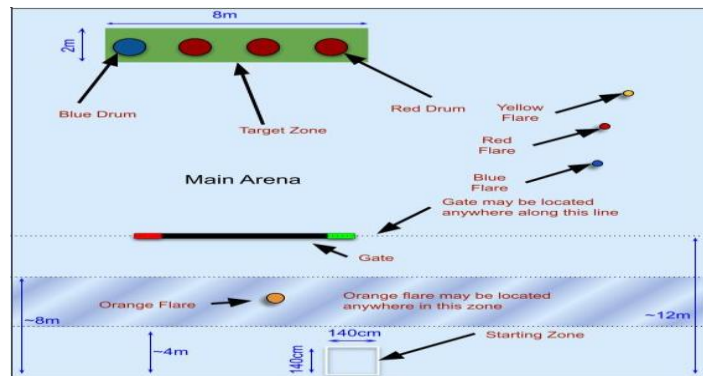
Pada Gambar 3.13 merupakan Perancangan *Hardware* yang digunakan pada Tugas Akhir ini. Terdapat beberapa sensor seperti kamera (*webcam*) yang berfungsi sebagai perekam objek dan mengirimkannya ke mini PC, *Pressure* sensor berfungsi untuk mempertahankan kedalaman AUR dan *Attitude angle* Sensor berfungsi sebagai pengukur dan mempertahankan sudut arah gerak AUR . Ketikan tombol otomatis ditekan, maka AUR akan menyelam dibawah permukaan air serta mempertahankan kedalaman serta arah gerak manuver sesuai dengan perintah microcontroller kemudian mini PC mengolah citra yang dikirimkan oleh webcam dengan algoritma YOLOv8 untuk menentukan objek berupa *gate*, *flare*, dan *drum*.

Hasil deteksi objek akan dikirimkan ke *microcontroller* ESP32 dan kemudian aktuator akan dikendalikan oleh microcontroller. jika kamera mendeteksi adanya *flare orange* maka ESP32 akan mengendalikan aktuator *thruster* untuk maju menghindari *flare orange*, ketika mendeteksi *gate* maka ESP32 akan mengendalikan aktuator *thruster* untuk maju melewati *gate*, Pada saat AUR mendeteksi adanya *flare* merah, biru, dan kuning maka ESP32 akan mengendalikan aktuator *thruster* untuk menabrak dan menjatuhkan bola golf pada ketiga *flare* tersebut secara bergantian, Ketika mendeteksi adanya *drum* biru maka ESP32 akan mengendalikan *Push Pull Solenoid* untuk menjatuhkan bola pada *drum* dan juga mengambilnya bola yang jatuh pada *drum* menggunakan mekanisme *Ball Taker* menggunakan motor BLDC.

Pada penggunaan aktuator seperti *Thruster* T200 dan BLDC motor diperlukan adanya *driver* seperti *Electronic Speed Control* yang berguna untuk mengontrol kecepatan dan arah berputar. Perencanaan sistem adalah tahap yang bertujuan membuat gambaran visual agar diketahui bagaimana bentuk desain dan menetapkan kebutuhan agar tercukupinya komponen penunjang pada suatu sistem. Berikut ini adalah bentuk tahap desain dan perencanaan sistem.

3.4.3 Desain Lintasan

Dalam Tugas Akhir ini, diperlukan sebuah desain lintasan untuk menguji sistem deteksi pada AUR, maka lintasan akan dibuat sesuai dengan aturan SAUVC 2025. Adapun bentuk lintasan yang dapat dilihat pada gambar 3.14



Gambar 3. 14 Desain Lintasan SAUVC 2025
(sauvc.org/rulebook, 2025)

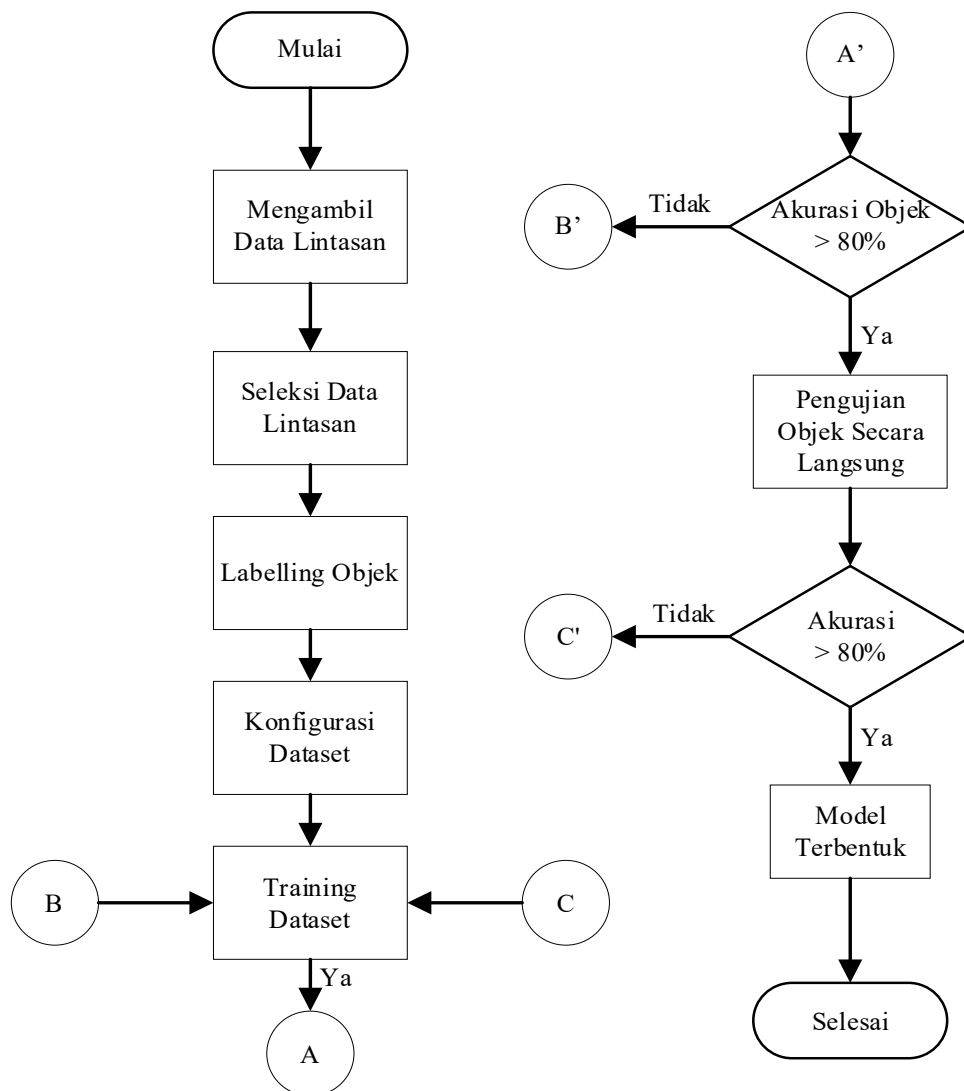
Pada gambar 3.10 terdapat empat zona yang harus diselesaikan pada lintasan tersebut. AUR dapat melanjutkan ke zona yang selanjut jika telah berhasil menyelesaikan zona sebelumnya. AUR dianggap finish atau selesai jika menyelesaikan zona satu hingga zona empat tanpa terlewatkan.

3.5 Dataset

Untuk mengetahui terbentuknya *dataset*, terdapat proses pengambilan *dataset* hingga model terbentuk diperlukan alur proses tahapan yang dapat dilihat pada *fLowchart* Gambar 3.15 Pada Tugas Akhir ini, dapat diketahui tentang proses awal yaitu pengambilan data yang diambil berupa video yang menampilkan bentuk lintasan dari *start* hingga *finish*. Data video yang diambil harus mencakup seluruh objek pada lintasan seperti *flare* orange, *gate*, *flare* biru, *flare* merah, *flare* kuning dan *drum* biru. Data lintasan yang telah diambil kemudian diubah menjadi bentuk gambar, kemudian diseleksi dengan menghapus data yang terindikasi blur dan kualitasnya kurang baik serta sebaiknya jelas memperlihatkan gambar objek yang ingin dideteksi. Data tersebut dikumpulkan satu file.

Data yang telah terkumpul, kemudian dilakukan labelling objek dengan menggunakan *platform* Yolo label. Pada proses *labelling* ini, objek pada gambar akan ditandai posisinya dengan lebih spesifik menggunakan koordinat dan diberikan *class* pada objek yang berbeda. Setelah dilakukan *labelling*, maka *dataset* telah terbentuk dan siap untuk dilakukan training *dataset*. *Training dataset* dilakukan dengan menggunakan *platform* google collabs yang berbasis *website*. Sebelum dilakukan *Training dataset* diperlukan konfigurasi yang benar agar sesuai

dengan kebutuhan, terdapat beberapa hal yang perlu dipersiapkan seperti pemilihan model versi YOLOv8, penyesuaian *epoch*, struktur *file* termasuk dataset dan *file* dengan format .yaml. *Training dataset* dilakukan untuk melatih model agar objek tersebut bisa dikenali menggunakan algoritma YOLOv8. Setelah dilakukan *training dataset*, akan didapatkan hasil berupa model yang telah terlatih berupa format .pt dan beberapa parameter lain. Kemudian dilakukan pengujian pada lintasan menggunakan data yang telah dilatih, jika hasil *confident* kurang dari 80% maka dataset akan dilatih ulang dan jika *confident* lebih dari 80% maka bisa dikatakan model terbentuk.



Gambar 3. 15 FLOWchart Proses Pembuatan Dataset

3.5.1 Penerapan Rumus Algoritma YOLO

Pada penerapannya, YOLOv8 memerlukan rumus yang digunakan melakukan deteksi objek dalam gambar secara efisien dan presisi oleh sebab itu memanfaatkan berbagai persamaan matematis guna mengonversi data gambar mentah menjadi prediksi yang mencakup lokasi objek, kategori, serta tingkat keyakinan terhadap keberadaan objek tersebut. Rumus yang digunakan pada Tugas Akhir ini antara lain

1. *Bounding box Prediction* (Prediksi kotak pembatas)

Bounding box prediction adalah menandai lokasi objek dalam gambar. Keuntungan dalam rumus ini yaitu meningkatkan kecepatan deteksi dan prediksi yang stabil dengan skala gambar yang berbeda. Rumus yang digunakan antara lain:

$$(b = (x, y, w, h)) \quad (3.1)$$

2. *Confusion Matrix*

Confusion matrix membandingkan hasil deteksi model oleh sistem dengan nilai deteksi yang sebenarnya, berguna untuk memahami jenis kesalahan yang dibuat oleh model, sehingga membantu dalam pengambilan keputusan untuk meningkatkan performa model. Rumus yang digunakan antara lain :

- Akurasi (*Accuracy*): Proporsi prediksi yang benar dari total prediksi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.2)$$

- Presisi (*Precision*): Proporsi prediksi positif yang benar dari semua prediksi positif.

$$Precision = \frac{TP}{TP + FP} \quad (3.3)$$

- *Recall* (Sensitivitas/*True Positive Rate*): Proporsi data positif yang berhasil diprediksi dengan benar.

$$Recall = \frac{TP}{TP + FN} \quad (3.4)$$

Keterangan:

- *True Positive* (TP)
- *True Negative* (TN)
- *False Positive* (FP)
- *False Negative* (FN)
- *F-1 Score* : rata-rata harmonik dari precision (ketepatan prediksi positif) dan recall (kemampuan model mendeteksi semua kasus positif)

$$F-1\ Score = (2 * Recall * Precision) / (Recall + Precision) \quad (3.5)$$

BAB 4

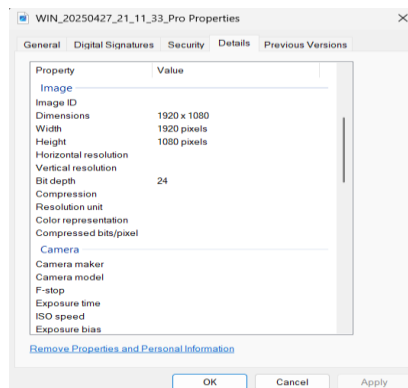
HASIL DAN PEMBAHASAN

4.1 Pengujian Sensor dan Aktuator

Pada Tugas Akhir ini terdapat proses pengujian sensor dan aktuator yang memiliki tujuan untuk mengetahui tingkat akurasi pada tiap komponen yang digunakan. Selain itu, terdapat tujuan lain yang digunakan untuk mengetahui besarnya presentase eror disetiap komponen yang digunakan. Berikut ini merupakan hasil dari pengujian dari sensor dan aktuator yang digunakan dalam Tugas Akhir.

4.1.1 Pengujian Sensor *Webcam*

Pada Tugas Akhir ini, *webcam* digunakan sebagai sensor utama pada *prototype* dan berfungsi sebagai pengambil citra secara *real-time*. Tipe webcam yang digunakan adalah *Logitech C390e* dengan resolusi 1920 x 1080 *Pixel* seperti yang bisa dilihat pada gambar 4.1



a)



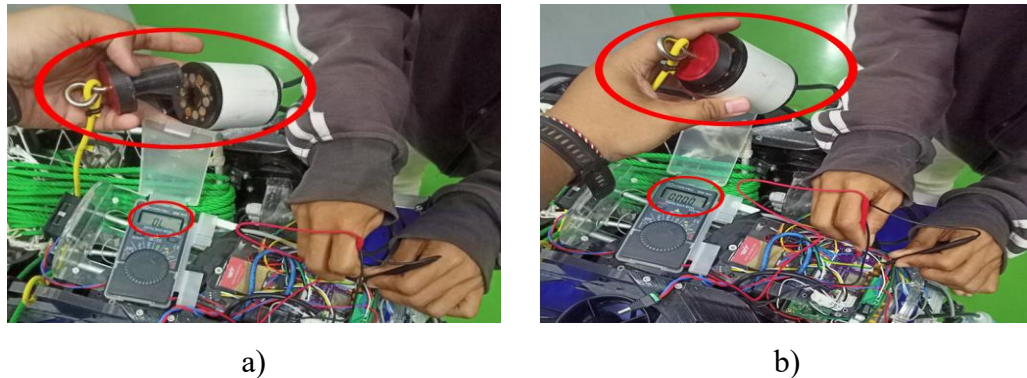
b)

Gambar 4. 1 a) Resolusi Citra Hasil Tangkapan *Webcam Logitech C390e* b) Indikator *Webcam* Terhubung Dengan PC atau Laptop

Bisa dilihat pada Gambar 4.1 a, merupakan spesifikasi hasil tangkapan *Logitech C390e*, sementara Gambar 4.1 b, *Webcam Logitech C390e* memiliki lampu indikator yang jika kondisi menyala atau terhubung dengan PC maupun Laptop, Lampu indikator akan menyala putih disisi kanan dan kiri *Webcam*.

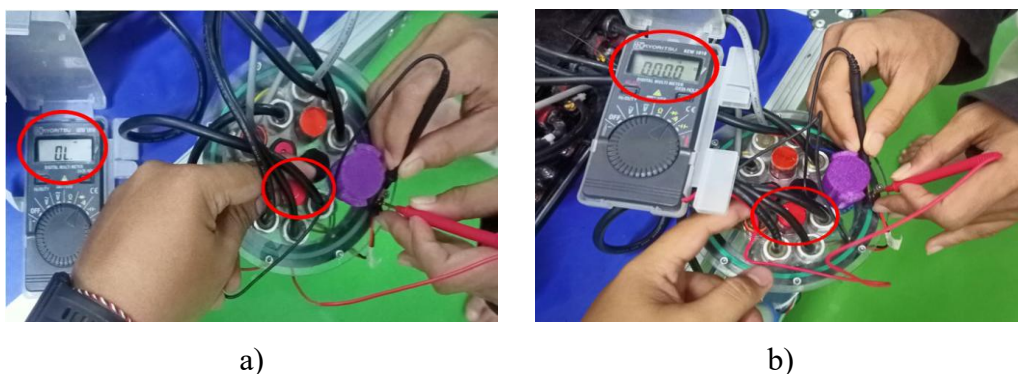
4.1.2 Pengujian Sensor *Switch*

Pada Tugas Akhir ini, *switch* digunakan sebagai *Input* untuk *trigger* menjalankan program, me-reset sistem, pemutus dan penghubung kelistrikan *prototype* dan komunikasi kabel. Pengujian *switch* dilakukan menggunakan avometer seperti Gambar 4.2.



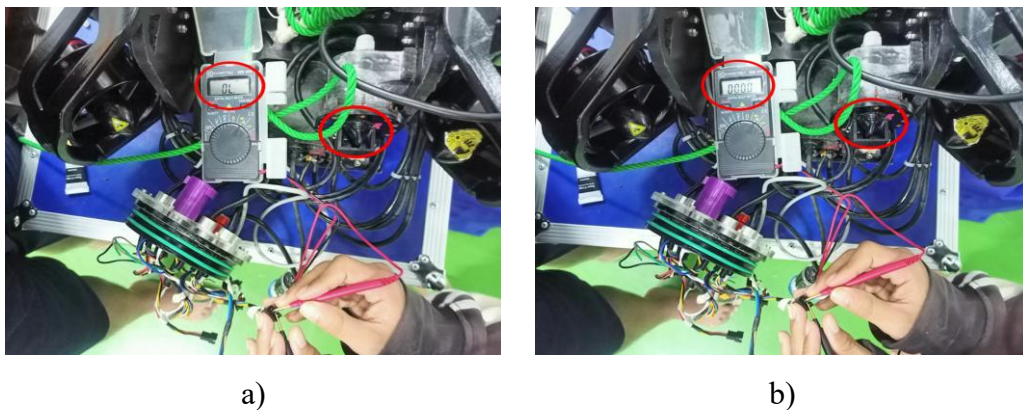
Gambar 4. 2 a) Kondisi *Magnetic Switch MC-38 Off* b) Kondisi *Magnetic Switch MC-38 On*

Percobaan dilakukan dengan *avometer* yang dihubungkan dengan kedua *pin magnetic switch* pada masing – masing probe. Kemudian menyetting *avometer* ke mode *buzzer* untuk mengecek koneksi, Gambar 4.2 a dan Gambar 4.2 b menunjukkan bahwa *magnetic switch* dalam kondisi baik, karena Ketika tidak didekatkan magnet maka nilai hambatan pada *display avometer* tidak muncul (*over load*) dan *buzzer* tidak berbunyi, sedangkan Ketika *magnetic switch* didekatkan magnet maka terdapat nilai hambatan pada *display avometer* dan *buzzer* akan berbunyi.



Gambar 4. 3 a) Kondisi *M10 Underwater Switch* Sedang Tidak Ditekan b) Kondisi *M10 Underwater Switch* Sedang Ditekan

Seperti percobaan sebelumnya, percobaan dilakukan dengan *avometer* yang dihubungkan dengan kedua *pin Underwater Switch* pada masing – masing probe. Kemudian menyetting *avometer* ke mode *buzzer* untuk mengecek koneksi, Gambar 4.3 a dan Gambar 4.3 b menunjukkan bahwa *Underwater Switch* dalam kondisi baik, karena Ketika tidak ditekan maka nilai hambatan pada *display avometer* tidak muncul (*over load*) dan *buzzer* tidak berbunyi, sedangkan Ketika *Underwater Switch* ditekan maka terdapat nilai hambatan pada *display avometer* dan *buzzer* akan berbunyi.

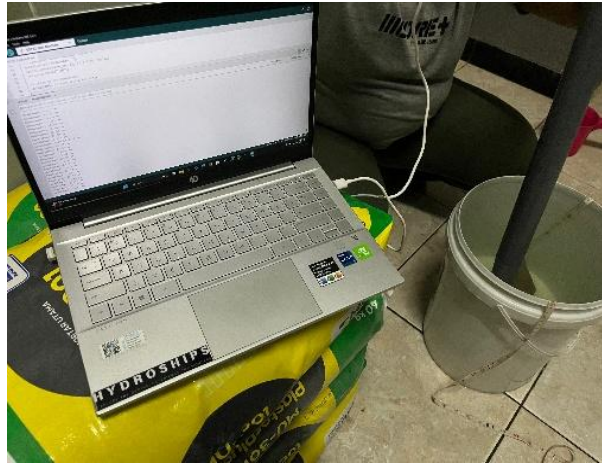


Gambar 4. 4 a) Kondisi M10 *Toggle Switch* MTS-102 Sedang Tidak Ditekan b) Kondisi M10 *Toggle Switch* MTS-102 Sedang Ditekan

Seperti percobaan sebelumnya, percobaan dilakukan dengan *avometer* yang dihubungkan dengan kedua *pin Toggle Switch* pada masing – masing probe. Kemudian menyetting *avometer* ke mode *buzzer* untuk mengecek koneksi, Gambar 4.7 dan Gambar 4.8 menunjukkan bahwa *Toggle Switch* dalam kondisi baik, karena Ketika tidak ditekan maka nilai hambatan pada *display avometer* tidak muncul (*over load*) dan *buzzer* tidak berbunyi, sedangkan Ketika *Toggle Switch* ditekan maka terdapat nilai hambatan pada *display avometer* dan *buzzer* akan berbunyi.

4.1.3 Pengujian *Pressure Sensor*

Pada Tugas Akhir ini, *Pressure sensor* digunakan sebagai pengukur kedalaman pada *prototype*, dimana ketika kedalaman sudah tercapai maka sensor ini digunakan untuk mempertahankan kedalaman *prototype*. Tipe *pressure sensor* digunakan adalah MS5837-30BA. Pengujian *pressure sensor* MS5837-30BA dapat dilihat pada Gambar 4.5 dan data hasil pengujian dapat dilihat pada Tabel 4.1.



Gambar 4. 5 Pengujian *Pressure* Sensor MS5837-30BA

Tabel 4. 1 Data Hasil Pengujian *Pressure* Sensor MS5837-30BA

Pengujian	Jarak Pengujian (mBar)	Pembacaan Sensor (mBar)	Nilai Error (%)
1	0	0	0
2	2,942	3,413	16
3	5,884	6,423	9,167
4	8,826	9,032	2,333
5	11,768	12,641	7,417
6	14,710	15,249	3,667
7	17,652	17,750	0,556
8	20,594	20,859	1,286
9	23,536	24,271	3,125
10	26,478	27,184	2,667
Rata - Rata Error			4,622

Setelah dilakukan pengujian pada *pressure* sensor MS5837-30BA terhadap alat ukur meteran dapat disimpulkan nilai rata – rata pada hasil pembacaan sensor hanya 4,622% dan dapat disimpulkan pembacaan sensor cukup akurat.

4.1.4 Pengujian *Attitude Angle* Sensor

Pada Tugas Akhir ini, *Attitude angle* sensor digunakan sebagai pengukur sudut kemiringan dari pergerakan AUR saat dibawah air. *Attitude angle* sensor yang digunakan adalah HWT905. Sensor ini akan dihubungkan ke microcontroller dan data yang dikirimkan berupa data 11 *byte* per parameter. Pengujian *attitude angle* sensor dapat dilihat pada Gambar 4.6



Gambar 4. 6 Perhitungan Sudut HWT905 *Attitude Angle* Sensor dengan Busur

Tabel 4. 2 Data Hasil Pengujian Sumbu X HWT905

Pengujian	Sudut Pengujian (°)	Pembacaan Sensor (°)	Nilai Akurasi (%)	Nilai Error (%)
1	0	0	0	0
2	10	10,14	98,6	1,4
3	20	20,41	97,95	2,05
4	30	30,35	98,83	1,17
5	40	40,52	98,7	1,3
6	50	50,27	99,46	0,54
7	60	60,08	99,87	0,13
8	70	70,03	99,6	0,04
9	80	80,9	98,87	1,13
10	90	90,3	99,67	0,33
Rata - Rata Nilai Sumbu X			99,81	0,81

Pengujian Sudut HWT905 *Attitude Angle* Sensor dilakukan dengan busur derajat dan diberikan penanda pada atas sensor. Setelah dilakukan pengujian pada sumbu X, dilihat pada Tabel 4.2 dapat disimpulkan nilai rata - rata akurasi 99,81% dan error 0,81%.

Tabel 4. 3 Data Hasil Pengujian Sumbu Y HWT905

Pengujian	Sudut Pengujian (°)	Pembacaan Sensor (°)	Nilai Akurasi (%)	Nilai Error (%)
1	0	0	0	0
2	10	10,15	98,5	1,5
3	20	20,1	99,5	0,5
4	30	30,37	98,77	1,23

Pengujian	Sudut Pengujian (°)	Pembacaan Sensor (°)	Nilai Akurasi (%)	Nilai Error (%)
5	40	40,23	99,43	0,57
6	50	50,18	99,64	0,36
7	60	60,19	99,68	0,32
8	70	70,42	99,4	0,6
9	80	80,53	99,34	0,66
10	90	90,21	99,77	0,23
Rata – Rata Nilai Sumbu Y			99,4	0,6

Pengujian Sudut HWT905 *Attitude Angle* Sensor dilakukan dengan busur derajat dan diberikan penanda pada atas sensor. Setelah dilakukan pengujian pada sumbu Y, dilihat pada Tabel 4.3 dapat disimpulkan nilai rata - rata akurasi 99,4% dan error 0,6%.

Tabel 4. 4 Data Hasil Pengujian Sumbu Z HWT905

Pengujian	Sudut Pengujian (°)	Pembacaan Sensor (°)	Nilai Akurasi (%)	Nilai Error (%)
1	0	0	0	0
2	10	10,34	96,6	3,4
3	20	20,27	98,65	1,35
4	30	30,13	99,57	0,43
5	40	40,37	99,08	0,92
6	50	50,26	99,48	0,52
7	60	60,52	99,13	0,87
8	70	70,29	99,59	0,41
9	80	80,44	99,45	0,55
10	90	90,78	99,13	0,87
Rata - Rata Error Sumbu Z			99,07	0,93

Pengujian Sudut HWT905 *Attitude Angle* Sensor dilakukan dengan busur derajat dan diberikan penanda pada atas sensor. Setelah dilakukan pengujian pada sumbu Z, dilihat pada Tabel 4.4 dapat disimpulkan nilai rata - rata akurasi 99,07% dan error 0,93%.

$$\begin{aligned}
 \text{Total rata-rata nilai akurasi} &= \frac{\text{Keseluruhan rata-rata nilai sumbu X,Y,Z}}{3} \\
 &= \frac{99,81+99,4+99,07}{3} = 99,42\%
 \end{aligned}$$

$$\text{Total rata-rata nilai error} = \frac{\text{Keseluruhan rata-rata nilai sumbu X,Y,Z}}{3}$$

$$= \frac{0,81+0,6+0,93}{3} = 0,78\%$$

4.1.5 Pengujian Aktuator Motor *Thruster*

Pada Tugas Akhir ini, Motor *thruster* digunakan sebagai penggerak utama sehingga AUR dapat bermanuver menggunakan gerakan 6 DOF(Degree Of Freedom) yang memungkinkan AUR untuk berguling, menunduk dan berputar dibawah air. Pergerakan motor thruster dikendalikan oleh nilai PWM dari 1100 us sampai 1900 us. Kecepatan motor akan diukur menggunakan *tachometer*. Motor thruster menggunakan tegangan 16V. Proses hasil pengukuran dapat dilihat pada Gambar 4.7 dan Tabel 4.5



Gambar 4. 7 Pengujian Kecepatan Motor *Thruster* Menggunakan *Tachometer*

Tabel 4. 5 Data Pengukuran Kecepatan Motor *Thruster* Menggunakan *Tachometer*

PWM (μ s)	Motor 1 (RPM)	Motor 2 (RPM)	Motor 3 (RPM)	Motor 4 (RPM)	Motor 5 (RPM)	Motor 6 (RPM)	Motor 7 (RPM)	Motor 8 (RPM)
1100	6429	6421	6444	6438	6432	6447	6449	6452
1200	5048	5039	5043	5049	5051	5056	5047	5042
1300	3621	3628	3624	3633	3622	3635	3629	3619
1400	1606	1602	1614	1618	1625	1608	1627	1615
1500	0	0	0	0	0	0	0	0
1600	1609	1606	1610	1613	1612	1608	1618	1611
1700	3623	3627	3635	3631	3621	3639	3630	3641

PWM (μ s)	Motor 1 (RPM)	Motor 2 (RPM)	Motor 3 (RPM)	Motor 4 (RPM)	Motor 5 (RPM)	Motor 6 (RPM)	Motor 7 (RPM)	Motor 8 (RPM)
1800	5042	5038	5044	5039	5047	5029	5042	5048
1900	6424	6430	6436	6423	6428	6411	6439	6433

Dapat dilihat pada Tabel 4.5 merupakan hasil pengukuran ke delapan *motor thruster* BLDC ROVMAKER T200 dengan variasi *Input* PWM menggunakan microcontroller ESP32 yang digunakan pada sistem penggerak AUR. Hasil pengujian pada Tabel 4.5 didapatkan nilai RPM maksimal 6439 dengan menggunakan variable *Input* PWM sebesar 1900 μ s.

4.1.6 Pengujian Aktuator BLDC Motor



Gambar 4. 8 Proses Pengukuran Kecepatan BLDC Motor Menggunakan Tachometer

Pada Tugas Akhir ini, BLDC Motor digunakan untuk mengambil bola pada saat bola berada dalam drum. Tipe BLDC Motor yang digunakan adalah F2429 yang menggunakan tegangan 12V. Putaran BLDC Motor dikendalikan oleh nilai PWM dari *microcontroller*. Pergerakan BLDC motor dikendalikan oleh nilai PWM dari 1100 us sampai 1900 us. Proses dan Hasil pengukuran dapat dilihat pada Gambar 4.8 dan Tabel 4.6

Tabel 4. 6 Data Pengukuran Kecepatan BLDC Motor Menggunakan *Tachometer*

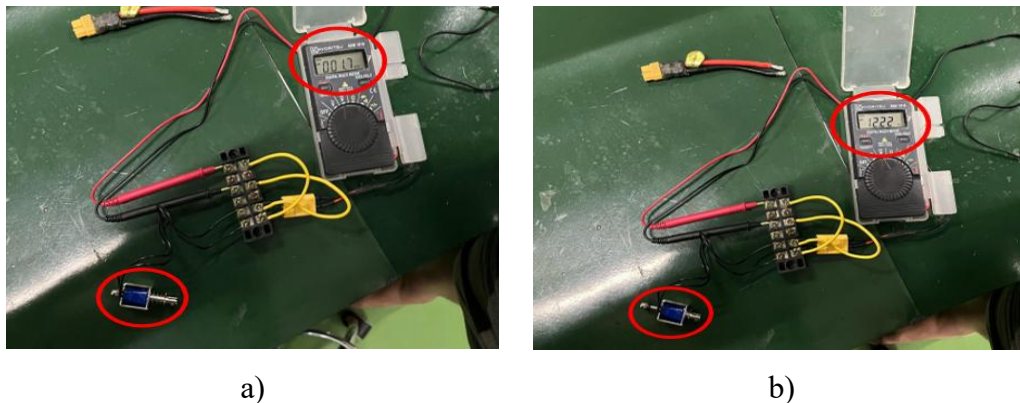
PWM (μ s)	BLDC Motor (RPM)
1100	2857
1200	10977
1300	8269

PWM (μs)	BLDC Motor (RPM)
1400	5626
1500	0
1600	6100
1700	8750
1800	11486
1900	2934

Dapat dilihat pada Tabel 4.6 merupakan hasil pengukuran BLDC Motor dengan variasi *Input* PWM menggunakan microcontroller ESP32 yang digunakan pada sistem penggerak AUR. Hasil pengujian pada Tabel 4.6 didapatkan nilai RPM maksimal 1189 dengan menggunakan variable *Input* PWM sebesar 1900 μs .

4.1.7 Pengujian Aktuator *Push Pull Solenoid*

Pada Tugas Akhir ini, *Push pull solenoid* digunakan untuk menjatuhkan bola kedalam drum. Tipe *push pull solenoid* yang digunakan adalah JF-0826B yang menggunakan tegangan 12V dengan *system switching* menggunakan relay yang di *switching* menggunakan sinyal digital pada *microcontroller*. Pengujian *push pull solenoid* JF-0826B dapat dilihat pada Gambar 4.9.



Gambar 4. 9 a) Kondisi *Push Pull Solenoid* JF-0826B Tidak Menerima Tegangan 12V b) Kondisi *Push Pull Solenoid* JF-0826B Menerima Tegangan 12V

Percobaan dilakukan dengan *avometer* yang dihubungkan dengan kedua *pin Push Pull Solenoid* JF-0826B pada masing – masing probe. Kemudian menyetting *avometer* ke mode VDC untuk mengecek tegangan, Gambar 4.9 a dan Gambar 4.9 b menunjukkan bahwa *Push Pull Solenoid* dalam kondisi baik, karena Ketika tidak

ada tegangan maka nilai voltase pada *display avometer* tidak muncul (*mendekati 0*) dan solenoid tidak bergerak maju sedangkan Ketika *Push Pull Solenoid* ada tegangan maka terdapat nilai voltase pada *display avometer* dan batang *solenoid* bergerak maju.

4.2 Pembuatan Sistem Mekanik

Perancangan pembuatan mekanik pada Tugas Akhir ini menggunakan berbagai macam bahan seperti *HDPE*, *filament PLA+ (3D printer)*, alumunium, dan akrilik sebagai kerangka, *ballast* dan bagian bagian AUR yang lainnya. Berikut merupakan beberapa proses pembuatan sistem mekanik pada Tugas Akhir ini.

4.2.1 Pembuatan Kerangka AUR

Bahan yang digunakan dalam pembuatan kerangka AUR adalah HDPE dengan ketebalan 10 mm. Sebelumnya kerangka AUR didesain terlebih dahulu kemudian HDPE akan dipotong menggunakan mesin CNC, Kemudian masing masing part akan disatukan untuk menjadi satu bagian yang utuh. Potongan dan penyatuan part dapat dilihat pada Gambar 4.10.



Gambar 4. 10 Hasil CNC Rangka AUR

Dapat dilihat pada Gambar 4.10 berupa hasil CNC rangka AUR, hasil CNC ini akan dipisahkan pada menurut bagiannya masing-masing. Pemisahan dilakukan dengan menggunakan gerinda untuk memisahkan bagian yang tidak dibutuhkan.

Kemudian dilakukan proses *countersink* pada lubang baut yang berfungsi agar kepala baut bisa rata dengan permukaan HDPE.



Gambar 4. 11 Hasil Penyatuan Rangka AUR

Pada pembuatan kerangka terdapat beberapa bagian seperti base plate, side frame dan mounting bracket. Semua bagian dirakit dan disambungkan menggunakan baut *stainless steel*. Pemilihan baut menggunakan bahan *stainless steel* berfungsi agar tidak terjadi korosi terhadap sambungan pada tiap bagian yang tersambung . Penggunaan bahan HDPE membuat konstruksi rangka AUR yang kokoh sehingga ketika mendapatkan benturan keras seperti terkena objek pada lintasan dan dinding kolam tetap tahan.

4.2.2 Pembuatan Ballast

Proses pembuatan *ballast* AUR dilakukan dengan menyatukan beberapa part yang telah terpecah. Bahan yang digunakan untuk pembuatan ballast adalah filament PLA+ dengan ketebalan 5 mm dan kepadatan struktur 30%. Proses pembuatan ballast AUR dapat dilihat pada Gambar 4.12



Gambar 4. 12 Proses Pembuatan Ballast

Dapat dilihat Gambar 4.12 proses pembuatan *ballast* dilakukan dengan menggunakan 3D print yang memakan waktu dari 8 hingga 14 jam untuk tiap bagian yang berbeda. Terdapat empat bagian dalam proses pembuatan *ballast* ini.



Gambar 4. 13 Hasil Pembuatan Ballast

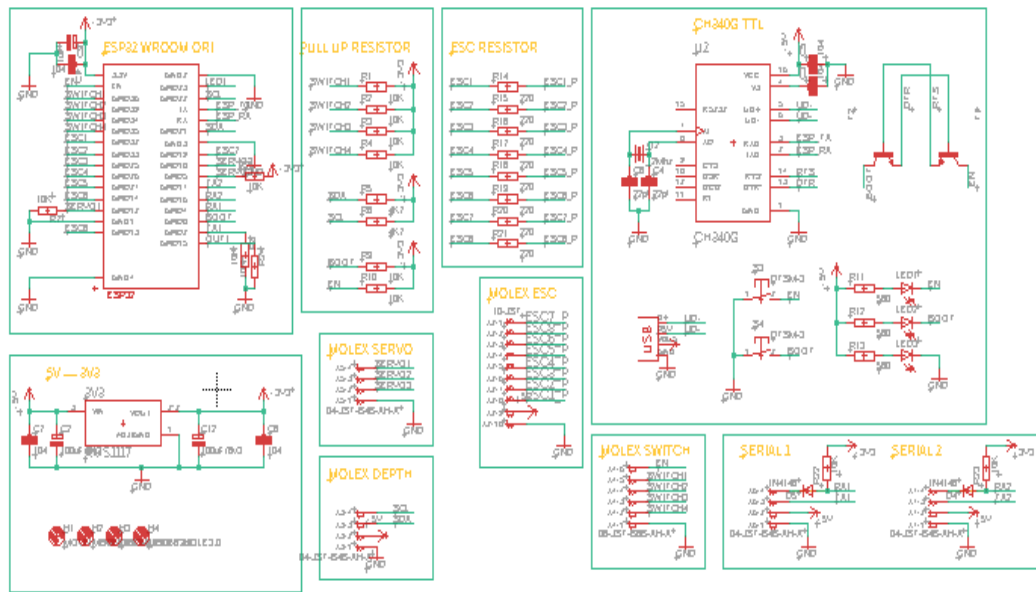
Dapat dilihat pada Gambar 4.13 merupakan hasil pembuatan *ballast*, Sebelum terbentuknya hasil tersebut perlu dilakukan penyatuan menggunakan Lem CA (*Cyanoacrylate*) pada 4 bagian yang telah dicetak. Setelah *ballast* disatukan dengan baik dilanjutkan melapisi dengan serat fiber dan ditambahkan bahan pengikat yaitu resin. Setelah resin kering dilanjutkan proses pelapisan dengan *polyester putty* dan pengamplasan. Proses pelapisan membutuhkan waktu 3 hari dan terakhir dilapisi dengan cat.

4.3 Pembuatan dan Perakitan Sistem Elektrik

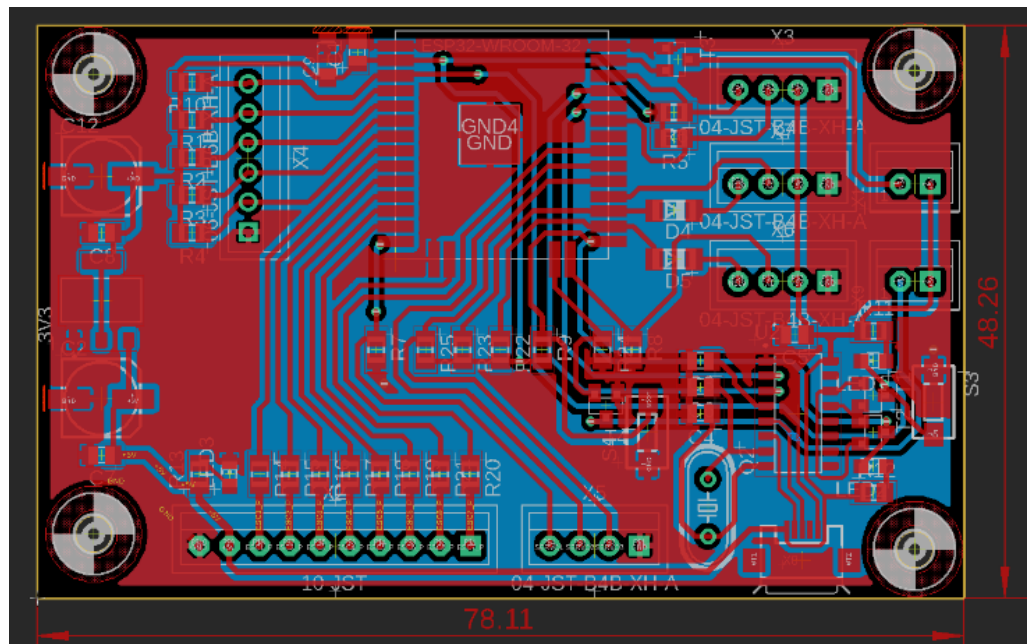
Pada pembuatan AUR diperlukan sistem elektrik yang berfungsi sebagai sarana penghubung antara pengolah data dari sensor dan kemudian di jadikan *Output* untuk aktuator. Pada Tugas Akhir ini, sistem elektrik seperti sensor, aktuator dan sistem suplai daya diintegrasikan pada sebuah mainboard berupa Printed Circuit Board (PCB). Berikut merupakan pembuatan dan perakitan sistem elektrik pada Tugas Akhir ini.

4.3.1 Pembuatan dan Perakitan Sistem Mainboard

Pembuatan sistem mainboard diawali dengan merancang sebuah desain sirkuit untuk menghubungkan kontroler utama yaitu ESP32 dengan beberapa komponen pendukung yang diperlukan untuk membentuk sistem yang terintegrasi pada prototype AUR. Desain sirkuit mainboard pada AUR dapat dilihat pada Gambar 4.14 dan Gambar 4.15

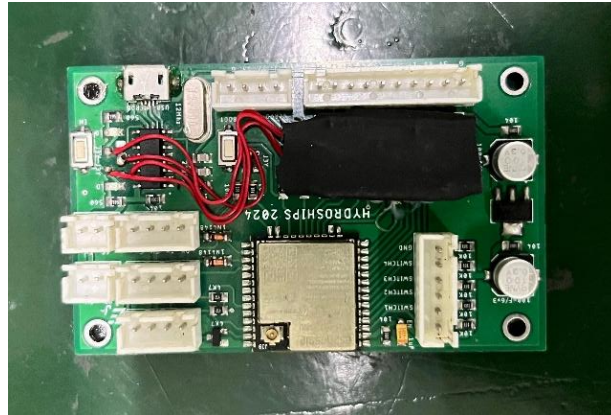


Pembuatan rangkaian skematik pada sistem *mainboard* perlu menggunakan komponen kecil atau *Surface-Mount Device* (SMD). Hal tersebut perlu dilakukan karena ukuran ruang pada enclosure utama yang minim sehingga pemilihan komponen tersebut harus dilakukan.



Gambar 4. 15 Hasil *Routing* PCB Sistem *Mainboard*

Setelah dilakukan proses *routing* pada *Mainboard*, maka akan dilakukan proses pencetakan board yang telah di rancang. Kemudian dilakukan proses perakitan komponen utama dan juga komponen pendukungnya pada *Mainboard*.

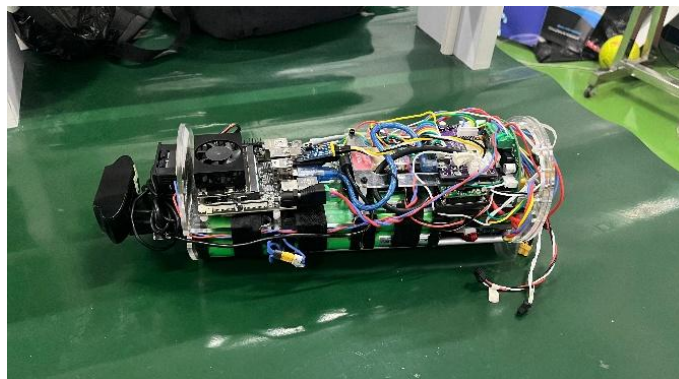


Gambar 4. 16 Hasil Cetak *Mainboard* Beserta Komponen Pendukung

Gambar 4.16 merupakan hasil cetak PCB dan hasil perakitan *mainboard* beserta komponen pendukungnya. *Mainboard* yang dibuat berbasis *microcontroller* ESP32 dan satu paket dengan uploadernya.

4.3.2 Instalasi Sistem Elektrik pada AUR

Pada sistem elektrik pada *prototype* AUR, Instalasi merupakan langkah yang paling krusial karena memerlukan keahlian dan ketelitian untuk memastikan optimalnya kinerja sistem dan meminimalisir terjadinya *short circuit*. Adapun beberapa proses instalasi elektrik seperti wiring electric menggunakan *electric tray* dan perakitan baterai sebagai suplai daya AUR. Bentuk instalasi elektrik pada AUR dapat dilihat pada Gambar 4.17 dan 4.18.



Gambar 4. 17 *Wiring Electric* pada *Electric Tray*

Pada Gambar 4.17 terdapat tiga bagian *electric tray* yaitu bagian atas, bagian bawah dan bagian depan. Pada bagian atas terdapat Mini PC yang diletakkan dibagian didepan *electric tray*, Sensor HWT905 diletakkan pada bagian Tengah, *mainboard* terletak pada belakang dan dibawahnya terletak *Interface Board*. Pada bagian bawah terdapat baterai tiga *cell* yang berjumlah 36 buah berkapasitas 36.000 mAh serta memiliki konfigurasi 12 paralel dan 3 seri yang hampir memenuhi ruang *electric tray*, dibagian belakang terdapat Battery Management System (BMS) dan empat relay yang berfungsi menghubungkan suplai daya tiga *cell* menuju seluruh system. Bagian depan terdapat *Webcam* Logitech C390e dan volt meter.

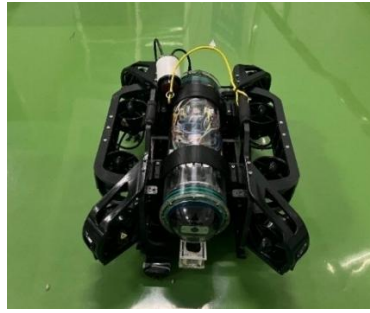


Gambar 4. 18 *Wiring Electric* pada *Enclosure Bawah*

Pada Gambar 4.18 merupakan hasil *wiring electric* pada *enclosure* bawah. Terdapat 8 ESC yang dihubungkan pada tiap kabel *thruster*, satu relay yang digunakan untuk menyalurkan suplai daya ke ESC, dan 2 shaft baterai yang dirangkai secara paralel, menggunakan 16 baterai pada 1 shaft. Perakitan 1 shaft baterai dibuat menjadi 4 *cell* serta berkonfigurasi 4 paralel dan 4 seri sehingga didapatkan tegangan 16V serta berkapasitas baterai sebesar 12.000 mAh. Pada *prototype* AUR ini menggunakan dua buah baterai *thruster* yang dirangkai secara paralel sehingga didapatkan kapasitas total sebesar 24.000 mAh.

4.3.3 Perakitan Seluruh Bagian AUR

Ketika seluruh bagian AUR telah dibuat maka tahapan terakhir adalah perakitan seluruh bagian AUR. Pada tahap ini seluruh bagian AUR dirakit dan disatukan menjadi satu *prototype* Hasil perakitan dapat dilihat pada Gambar 4.19 dan Gambar 4.20.



Gambar 4. 19 Perakitan Seluruh Sistem Perangkat Keras dan Penggerak AUR

Perakitan *enclosure* utama pada AUR dilakukan dengan menghubungkan kabel menggunakan konektor tertentu sehingga memudahkan dalam proses pembongkaran dan perakitan, penggunaan konektor juga dilakukan pada *enclosure* bawah.



Gambar 4. 20 Perakitan Sistem Ballast Tetap pada *Prototype AUR*

Setelah dilakukan perakitan seluruh perangkat terhadap *prototype AUR* maka dilakukan pengujian terhadap kedekatan *enclosure* utama dan *enclosure* bawah agar dapat meminimalisir terjadinya kebocoran terhadap *prototype AUR* sebelum digunakan



Gambar 4. 21 Proses Perendaman *Prototype AUR*

Dapat dilihat pada Gambar 4.21 pengujian kedap air dilakukan dengan cara memberikan pemberat pada AUR dan menenggalamkan di kedalaman 1 meter selama 1 jam. Setelah itu, akan dilakukan pengecekan kebocoran, ketika enclosure electric dan enclosure bawah pada prototype AUR mengalami kebocoran lebih dari 50 mL, maka kedua enclosure akan dilakukan penambalan pada titik yang menyebabkan kebocoran dan apabila kebocoran kurang dari 50 mL maka prototype dinyatakan layak untuk digunakan.

4.4 Pembuatan Dataset

Dalam pembuatan *dataset* terdapat beberapa proses yang harus dilakukan. Pertama ialah proses pengulang *dataset* dan yang kedua ialah proses anotasi *dataset*. Berikut adalah penjelasan hasil pembuatan data set.

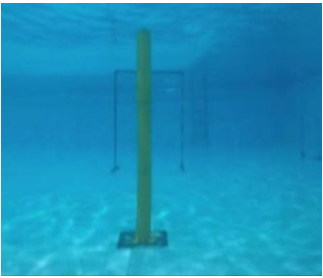
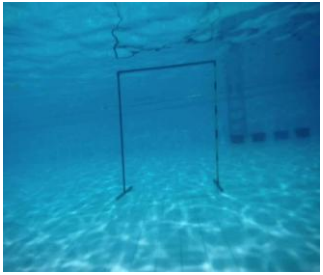
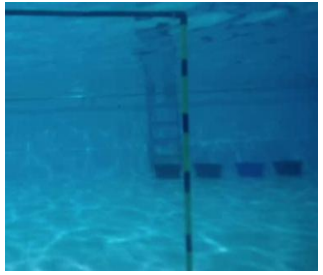


4.4.1 Pengumpulan *dataset*


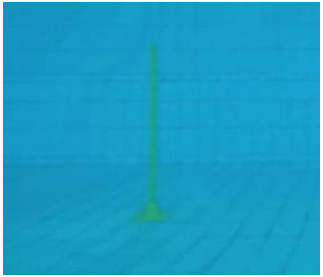

Proses dalam membuat dataset adalah proses pengumpulan *dataset*. Proses ini dilakukan memvideo yang kemudian akan dipecah menjadi gambar. Objek yang diambil berupa *flare* orange, *gate*, *flare* biru, *flare* merah, *flare* kuning, dan *drum* biru pada lintasan yang telah dibuat. Pengumpulan ini dilakukan secara manual menggunakan *webcam logitech c930e* yang merupakan *webcam* sama yang digunakan pada AUR agar data yang diperoleh memiliki kualitas yang baik. Hasil dan proses pengambilan data dapat dilihat pada Gambar 4.27 dan Tabel 4.7.



Gambar 4. 22 Proses Pengambilan *Dataset*

Tabel 4. 7 Hasil Pengambilan Data Objek Lintasan SAUVC

No.	Gambar Dataset	Keterangan	Jumlah Dataset
1		<i>Obstacle orange</i>	349
2		<i>Gate utama</i>	169
3		<i>Gate kanan</i>	445
4		<i>Gate kiri</i>	372
5		<i>Flare biru</i>	603

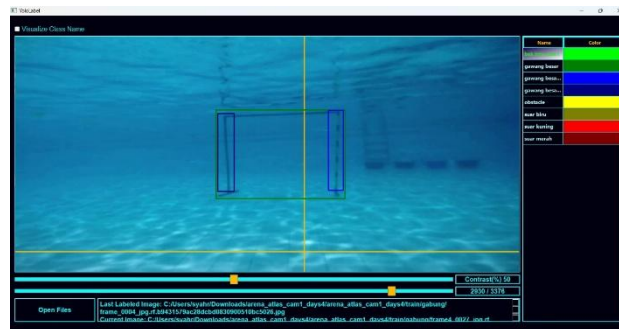
No.	Gambar Dataset	Keterangan	Jumlah Dataset
6		<i>Flare merah</i>	677
7		<i>Flare kuning</i>	466
8		<i>Drum Biru</i>	393
Jumlah Total Dataset			2075

Dataset objek pada lintasan SAUVC yang diambil dibawah air pada kondisi waktu malam. Hal tersebut dilakukan karena aturan dari pihak manajemen yang mengharuskan untuk memilih waktu yang sama pada setiap *trial*. Pengambilan *dataset* dilakukan sebanyak 3 kali dengan kondisi cahaya yang tetap. Dari proses pengambilan *dataset* tersebut didapatkan 10.000 gambar objek. Kemudian dataset tersebut akan dipilah menjadi 7.200 gambar yang akan digunakan sebagai data *training* dan validasi.

4.4.2 *Labelling Dataset*

Dataset yang sudah diperoleh tidak bisa langsung digunakan untuk *training* model YOLOv8. Diperlukan sebuah proses *labelling* yang bertujuan untuk memberikan tanda objek yang akan dideteksi. Tanda tersebut berupa *class* objek

dan posisi objek atau titik koordinat yang ditandai dengan *bounding box*. Proses *labelling dataset* dapat dilihat pada Gambar 4.23.

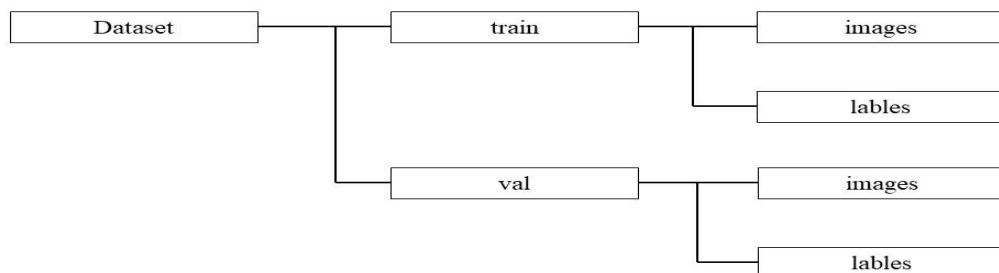


Gambar 4. 23 Proses *Labelling Dataset*

Setelah dilakukannya proses *labelling dataset*, didapatkan hasil berupa *file* yang berisi gambar objek dan anotasi objek dengan format *.txt* yang berisi tentang *class* dan letak masing-masing objek pada setiap gambar *dataset*. *File* ini digunakan sebagian acuan dalam proses *training* untuk menentukan objek mana yang akan dideteksi sehingga model dapat memahami karakteristik dan keunikan masing-masing objek.

4.4.3 *Split Dataset*

Setelah melakukan proses pemberian tanda objek yang akan dideteksi, terdapat tahapan terakhir dalam pembentukan *dataset* yaitu *split dataset*. *Split dataset* dilakukan dengan tujuan untuk membagi *dataset* menjadi dua bagian yaitu data *training* dan data validasi. Data *training* nantinya digunakan sebagai acuan dalam melatih bobot. Sedangkan data validasi digunakan memvalidasi hasil dari proses *training*. Adapun bentuk konfigurasi *split dataset* yang dapat dilihat pada Gambar 4.24.



Gambar 4. 24 Konfigurasi *Split Dataset*

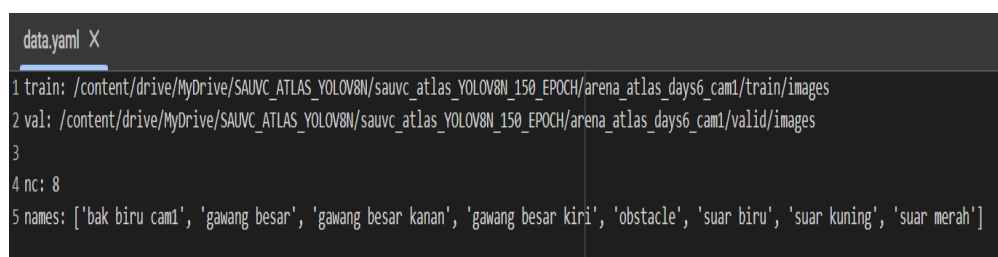
Pada Gambar 4.24 *split dataset* dibagi menjadi dua folder yang berisi train 80% dan validasi 20%. Masing-masing folder berisi dua folder yang berisi *images* dan *labels*. Folder *images* berisi seluruh gambar *dataset* sedangkan folder *labels* berisi file .txt hasil dari proses labeling *dataset*. Data yang digunakan sebagai proses *training* atau melatih bobot yaitu folder *train* dengan jumlah 5175 *dataset* dan data pada folder *validasi* akan digunakan untuk melakukan validasi bobot dengan jumlah 2075 *dataset*.

4.5 Training Model YOLOv8

Dalam proses pembentukan model diperlukan tahapan *training* model. Tahapan ini merupakan tahapan paling penting karena dapat menentukan kemampuan model dalam mendeteksi objek.

4.5.1 Pengaturan Parameter *Training*

Sebelum dilakukan *training dataset*, terdapat langkah awal yang harus dilakukan yaitu konfigurasi parameter *training*. Konfigurasi ini bertujuan untuk menentukan *dataset* mana yang akan digunakan serta kelas apa saja yang akan dideteksi. Konfigurasi ini dilakukan pada file dengan nama *data.yaml*. Isi dari *file* tersebut adalah alamat dari *dataset* dan juga kelas objek yang akan dideteksi. Berikut tampilan isi dari *file data.yaml* yang dapat dilihat pada Gambar 4.25.



```
data.yaml X
1 train: /content/drive/MyDrive/SAUVC_ATLAS_YOLOV8N/sauvc_atlas_YOLOV8N_150_EPOCH/arena_atlas_days6_cam1/train/images
2 val: /content/drive/MyDrive/SAUVC_ATLAS_YOLOV8N/sauvc_atlas_YOLOV8N_150_EPOCH/arena_atlas_days6_cam1/valid/images
3
4 nc: 8
5 names: ['bak biru cam1', 'gawang besar', 'gawang besar kanan', 'gawang besar kiri', 'obstacle', 'suar biru', 'suar kuning', 'suar merah']
```

Gambar 4. 25 *file data.yaml*

Selain pengaturan *dataset*, konfigurasi ini juga digunakan untuk mengatur parameter lain yaitu *epoch*, *batch*, dan varian bobot YOLOv8 yang akan digunakan. Berikut adalah parameter yang digunakan untuk *training* model YOLOv8 yang tersedia pada Tabel 4.8.

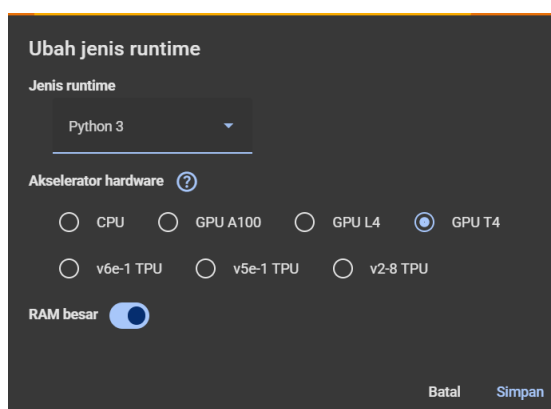
Tabel 4. 8 Parameter Training

Parameter	Keterangan
Epoch	150
Batch	16
Varian Model	YOLOv8n

Dari Konfigurasi tersebut, diketahui bahwa training YOLOv8 dilakukan sebanyak 150 *epoch*, dengan 16 batch (jumlah anggota kelompok sample) dan *training* dengan varian bobot YOLOv8n

4.5.2 Proses *Training*

Setelah dilakukan penetapan parameter training, dilanjutkan dengan proses *training dataset*. Proses ini menggunakan *platform google collabs* berbasis *cloud* yang berfungsi melatih suatu dataset yang digunakan untuk sistem deteksi YOLOv8. Adapun beberapa Langkah-langkah yang digunakan seperti:



Gambar 4. 26 *Setting Jenis Runtime*

Sesuai dengan Gambar 4.26 sebelum menggunakan *google collabs*, pengguna harus memilih jenis *runtime* yang cocok disesuaikan dengan kebutuhan. Penggunaan bahasa pemrograman *python 3* karena bahasa lebih mudah dipahami, akselerator *hardware* menggunakan GPU T4 karena membutuhkan performa lebih tinggi, dan menggunakan RAM besar untuk mencegah penggunaan memori RAM yang penuh. Setelah memilih jenis *runtime*, dilanjutkan dengan menghubungkan *runtime* dengan *google collabs*.


```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Gambar 4. 27 Menghungkan *File Google Drive*

Sesuai dengan Gambar 4.27 merupakan proses mehubungkan *file dataset* yang sudah diupload pada *google drive* pada *platform google collabs*. Proses ini paling umum digunakan untuk menggunakan *dataset* yang sudah melalui proses anotasi.

```
%cd /content/drive/MyDrive/sauvc_atlas_YOLOV8N_150_EPOCH
```

Gambar 4. 28 Merubah Direktori Penyimpanan

Sesuai dengan Gambar 4.28 merupakan proses mengubah direktori yang digunakan untuk menyimpan semua file hasil perintah dari script pada *google collabs*. Proses ini menguntungkan karena setelah hasil training dataset selesai, hasil tersebut akan otomatis tersimpan pada jalur *file* yang telah dipilih.

```
%pip install ultralytics
import ultralytics
ultralytics.checks()
```

Gambar 4. 29 *Install Pustaka Ultralytics*

Sesuai dengan Gambar 4.29 merupakan proses instalasi pustaka dari *ultralytics*. *Google collabs* tidak memiliki pustaka ini secara default, perintah ini memastikan pustaka diunduh dan diinstalasi ke lingkungan runtime. Setelah diinstalasi, Pustaka siap dan bisa digunakan.

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')
model.train(data='/content/drive/MyDrive/sauvc_atlas_YOLOV8N_150_EPOCH/arena_atlas_days6_cam1/data.yaml', epochs=150, batch=16)
```

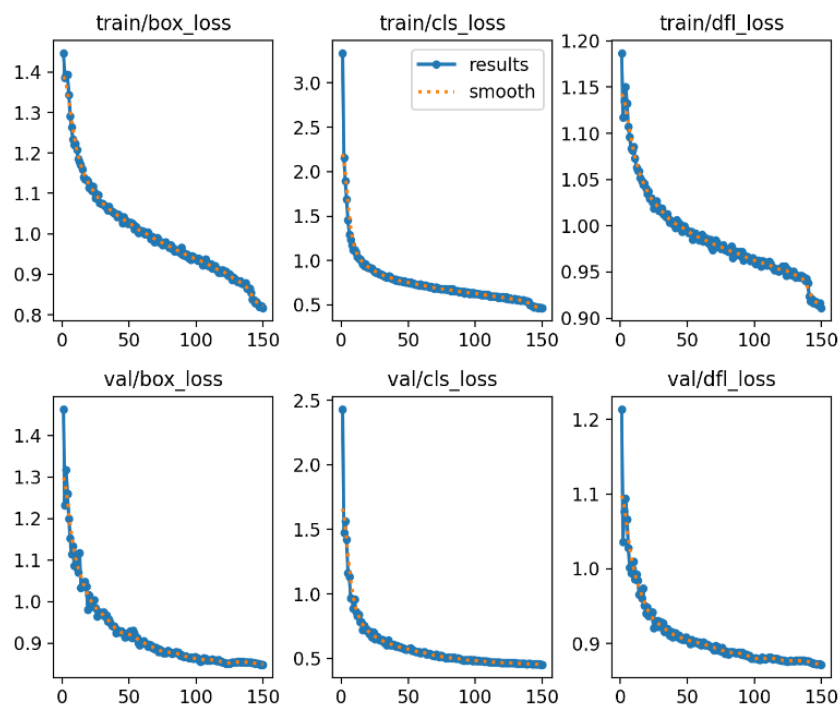
Gambar 4. 30 Kode *Training YOLOv8*

Sesuai dengan Gambar 4.30 merupakan kode *training* yang digunakan pada *google collabs*. Sebelum menjalankan *training*, pengguna harus menentukan model

YOLO, *epoch*, *batch* dan yang paling utama mengetahui Lokasi *file* dengan format .yaml. Setelah kode *training* tersebut dijalankan, maka pengguna perlu menunggu beberapa saat hingga *training* selesai dijalankan.

4.5.3 Hasil Training

Setelah dilakukan *training* dengan menggunakan 150 *epoch* dan membutuhkan waktu sebanyak 2 jam 27 menit. Hasil *training* model YOLOv8 disajikan pada Gambar 4.31.

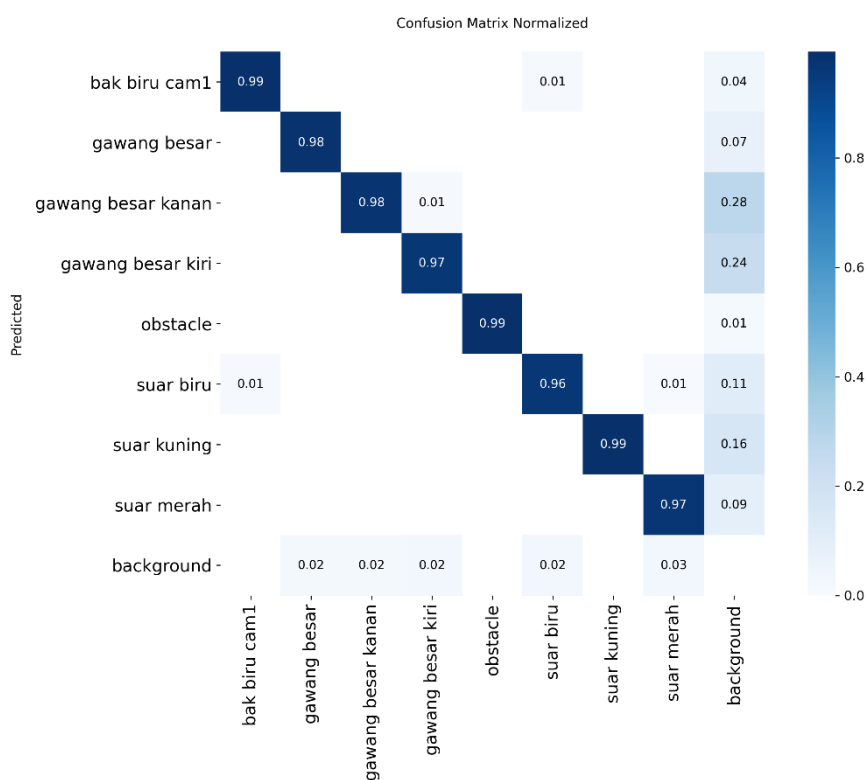


Gambar 4. 31 Hasil *Training* Model YOLOv8

Dari gambar tersebut dapat diketahui bahwa hasil *training* YOLOv8n memiliki tingkat *box_loss* mendekati 0.8 diakhir epoch yang menunjukkan bahwa model semakin presisi dalam menggambar *bounding box* terhadap objek yang dideteksi. *Loss* adalah tingkat kehilangan objek yang diprediksi atau tingkat kegagalan objek yang diprediksi atau kemampuan model dalam prediksi objek. Hasil *cls_loss* mendekati 0.5 yang menunjukkan model mampu mengklasifikasikan objek dengan lebih akurat seiring bertambahnya *epoch*. Hasil *dfl_loss* turun menjadi 0.9 menunjukkan bahwa model semakin presisi dalam merepresentasikan posisi *bounding box* secara detail.

4.5.4 Hasil Pengujian menggunakan Confusion Matrix

Sebelum dilakukan pengujian dengan cara mendeteksi langsung objek perlu dilakukan pengujian secara matematis menggunakan *Confusion Matrix*. Pengujian ini bertujuan untuk memprediksi kemampuan model dalam mendeteksi objek secara langsung, pengujian ini menggunakan data validasi dalam proses *training* sebagai data acuannya dimana terdapat 393 sampel objek *drum* biru, 169 sampel objek *gate* utama, 445 sampel objek *gate* kanan, 372 sampel objek *gate* kiri, 349 sampel objek *flare* orange, 603 sampel objek *flare* biru, 466 sampel objek *flare* kuning, dan 677 sampel objek *flare* merah. Berikut adalah *Confusion Matrix* performa model yolov5m yang telah di *training* di sajikan pada Gambar 4.32.



Gambar 4. 32 *Confusion Matrix* Hasil *Training* Model YOLOv8n

Pada Gambar 4.32 terdapat dua sisi yaitu *predicted* dan *true*. *Predicted* ialah hasil dari prediksi model pada data validasi, sedangkan *true* ialah label kebenaran, pada masing-masing sisi terdapat 9 kelas yaitu *drum* biru(bak biru cam1), *gate* utama(gawang besar), *gate* kanan(gawang besar kanan), *gate*(gawang besar kiri), *flare* orange(*obstacle*), *flare* biru (suar biru), *flare* merah(suar merah), *flare*

kuning(suar kuning), dan *background*. Kelas *background* kelas yang muncul akibat terjadinya kesalahan deteksi objek, semisal terdapat benda lain yang terdeteksi walaupun benda tersebut bukan dari anggota objek yang seharusnya dideteksi. Kelas *background* biasanya menjadi representasi dari kondisi FP (False Positif) dan FN (False Negatif). Setelah diperoleh *confusion matrix*, selanjutnya dilakukan perhitungan untuk *Accuracy*, *Precision*, *Recall* dan F-1 Score. Parameter tersebut digunakan untuk mengetahui performa model dalam memprediksi Berikut adalah hasil perhitungan *Accuracy*, *Precision*, *Recall* dan F-1 Score pada kelas *drum* biru.

$$TP = 389$$

$$TN = 6 + 11 = 17$$

$$FP = 166 + 435 + 1 + 348 + 4 + 1 + 1 + 3 + 8 + 9 + 1 = 977$$

$$FN = 3723 - (389 + 17 + 977) = 2340$$

Untuk mengetahui *accuracy* model dalam memprediksi *drum* biru dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (435 + 1912) / 3723$$

$$Accuracy = 73.9\%$$

selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 435 / (435 + 1292)$$

$$Precision = 28.5\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *drum* biru merah dihitung dengan persamaan 4.3.

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 389 / (389 + 17)$$

$$Recall = 95.8\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *drum* biru dihitung dengan persamaan 4.4.

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.252 * 0.838) / (0.252 + 0.838)$$

$$F - 1 \text{ Score} = 43.9\%$$

Untuk mengetahui performa model dalam memprediksi *gate utama*. Perhitungan yang sama dengan *gate* utama sebagai berikut.

$$TP = 435$$

$$TN = 3 + 81 = 84$$

$$FP = 389 + 166 + 1 + 361 + 348 + 4 + 1 + 1 + 3 + 8 + 9 + 1 = 1292$$

$$FN = 3723 - (435 + 84 + 1292) = 1912$$

Untuk mengetahui *accuracy* model dalam memprediksi *gate* utama dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (435 + 1912) / 3723$$

$$Accuracy = 63.1\%$$

selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 435 / (435 + 1292)$$

$$Precision = 25.2\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *gate* utama merah dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 435 / (435 + 84)$$

$$Recall = 83.8\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *gate* utama dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.252 * 0.838) / (0.252 + 0.838)$$

$$F - 1 \text{ Score} = 38.8\%$$

Untuk mengetahui performa model dalam memprediksi *gate* kanan. Perhitungan yang sama dengan *gate* kanan sebagai berikut.

$$TP = 361$$

$$TN = 70$$

$$FP = 389 + 166 + 435 + 1 + 348 + 4 + 1 + 1 + 3 + 8 + 9 + 1 = 1327$$

$$FN = 3723 - (361 + 70 + 1327) = 1965$$

Untuk mengetahui *accuracy* model dalam memprediksi *gate* kanan dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (361 + 1965) / 3723$$

$$Accuracy = 63.5\%$$

selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 361 / (361 + 1327)$$

$$Precision = 21.4\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *gate* kanan merah dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 361 / (361 + 70)$$

$$Recall = 83.8\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *gate* kanan dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.214 * 0.838) / (0.214 + 0.838)$$

$$F - 1 \text{ Score} = 34.1\%$$

Untuk mengetahui performa model dalam memprediksi *gate* kiri. Perhitungan yang sama dengan *gate* kiri sebagai berikut.

$$TP = 348$$

$$TN = 2 + 4 = 6$$

$$FP = 389 + 166 + 435 + 1 + 361 + 4 + 1 + 1 + 3 + 8 + 9 + 1 = 1379$$

$$FN = 3723 - (348 + 6 + 1379) = 1990$$

Untuk mengetahui *accuracy* model dalam memprediksi *gate* kiri dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (348 + 1990) / 3723$$

$$Accuracy = 63.2\%$$

Selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 348 / (348 + 1379)$$

$$Precision = 20.1\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *gate* kiri merah dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 348 / (348 + 6)$$

$$Recall = 98.3\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *gate* kiri dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.201 * 0.983) / (0.201 + 0.983)$$

$$F - 1 \text{ Score} = 33.4\%$$

Untuk mengetahui performa model dalam memprediksi *flare* orange. Perhitungan yang sama dengan *flare* orange sebagai berikut.

$$TP = 582$$

$$TN = 4 + 33 = 37$$

$$FP = 389 + 166 + 435 + 1 + 361 + 348 + 2 + 4 + 1 + 1 + 3 + 8 + 9 + 1 \\ = 2329$$

$$FN = 3723 - (582 + 37 + 2329) = 775$$

Untuk mengetahui *accuracy* model dalam memprediksi *flare* orange dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (582 + 775) / 3723$$

$$Accuracy = 36.7\%$$

Dari perhitungan *accuracy* pada persamaan 4.1 nilai akurasi diperoleh dan memiliki tingkat akurasi di atas 36.7% maka selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 348 / (348 + 1379)$$

$$Precision = 20.1\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *flare* orange dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 582 / (582 + 2329)$$

$$Recall = 94\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *flare* orange dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.201 * 0.983) / (0.201 + 0.983)$$

$$F - 1 \text{ Score} = 33.4\%$$

Untuk mengetahui performa model dalam memprediksi *flare* biru. Perhitungan yang sama dengan *flare* orange sebagai berikut.

$$TP = 461$$

$$TN = 46$$

$$FP = 389 + 166 + 435 + 1 + 361 + 348 + 2 + 4 + 4 + 1 + 3 + 8 + 9 + 1 \\ = 2332$$

$$FN = 3723 - (461 + 46 + 2332) = 884$$

Untuk mengetahui *accuracy* model dalam memprediksi *flare* biru dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (461 + 884) / 3723$$

$$Accuracy = 36.3\%$$

selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 461 / (461 + 2332)$$

$$Precision = 16.5\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *flare* biru merah dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 461 / (461 + 46)$$

$$Recall = 90.9\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *flare* biru dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.165 * 0.909) / (0.165 + 0.909)$$

$$F - 1 \text{ Score} = 28\%$$

Untuk mengetahui performa model dalam memprediksi *flare* kuning. Perhitungan yang sama dengan *flare* kuning sebagai berikut.

$$TP = 656$$

$$TN = 26$$

$$FP = 389 + 166 + 435 + 1 + 361 + 348 + 2 + 4 + 4 + 1 + 1 + 3 + 8 + 10 \\ = 2333$$

$$FN = 3723 - (656 + 26 + 2333) = 708$$

Untuk mengetahui *accuracy* model dalam memprediksi *flare* kuning dilakukan perhitungan dengan persamaan 4.1.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$Accuracy = (656 + 708) / 3723$$

$$Accuracy = 36.7\%$$

selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$Precision = (TP) / (TP + FP) \quad (4.2)$$

$$Precision = 656 / (656 + 2333)$$

$$Precision = 21.9\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *flare* kuning merah dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 656 / (656 + 26)$$

$$Recall = 90.9\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *flare* kuning dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.219 * 0.962) / (0.219 + 0.962)$$

$$F - 1 \text{ Score} = 35.7\%$$

Untuk mengetahui performa model dalam memprediksi *flare* merah. Perhitungan yang sama dengan *flare* kuning sebagai berikut.

$$TP = 166$$

$$TN = 21$$

$$FP = 389 + 435 + 1 + 361 + 348 + 4 + 1 + 1 + 3 + 8 + 9 + 1 = 2162$$

$$FN = 3723 - (166 + 21 + 2162) = 1374$$

Untuk mengetahui *accuracy* model dalam memprediksi *flare* merah dilakukan perhitungan dengan persamaan 4.1.

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN) \quad (4.1)$$

$$\text{Accuracy} = (166 + 1374) / 3723$$

$$\text{Accuracy} = 41\%$$

Dari perhitungan *accuracy* pada persamaan 4.1 nilai akurasi diperoleh dan memiliki tingkat akurasi di atas 41 % maka selanjutnya dilakukan perhitungan nilai *precision* dengan persamaan 4.2.

$$\text{Precision} = (TP) / (TP + FP) \quad (4.2)$$

$$\text{Precision} = 166 / (166 + 2162)$$

$$\text{Precision} = 7.1\%$$

Selanjutnya untuk mengetahui nilai *recall* model dalam memprediksi *flare* merah merah dihitung dengan persamaan 4.3

$$Recall = TP / (TP + FN) \quad (4.3)$$

$$Recall = 166 / (166 + 2162)$$

$$Recall = 88.8\%$$

Setelah memperoleh nilai *accuracy*, *precision*, dan *recall* untuk mengetahui nilai *f1 score* model dalam memprediksi *flare* merah dihitung dengan persamaan 4.4

$$F - 1 \text{ Score} = 2 * (Recall * Precision) / (Recall + Precision) \quad (4.4)$$

$$F - 1 \text{ Score} = 2 * (0.071 * 0.888) / (0.071 + 0.888)$$

$$F - 1 \text{ Score} = 13.2\%$$

Untuk mempermudah dalam pembacaan data hasil pengujian menggunakan *Confusion Matrix* seluruh data dirangkum dan disajikan pada Tabel

Tabel 4. 9 Hasil Pengujian Model dengan *Confusion Matrix*

Objek	YOLOv8n	
Drum biru	TP	389
	TN	17
	FP	977
	FN	2340
	Perhitungan Manual	
	<i>Accuracy</i>	73.9%
	<i>Prescision</i>	28.5%
	<i>Recal</i>	95.8%
	<i>F-1 Score</i>	43.9%
Gate utama	TP	435
	TN	84
	FP	1292
	FN	1912
	Perhitungan Manual	
	<i>Accuracy</i>	63.1%
	<i>Prescision</i>	25.2%
	<i>Recal</i>	83.8%
	<i>F-1 Score</i>	38.8%

Objek	YOLOv8n	
Gate kanan	TP	361
	TN	70
	FP	1327
	FN	1965
	Perhitungan Manual	
	<i>Accuracy</i>	62.5%
	<i>Prescision</i>	21.4%
	<i>Recal</i>	83.8%
	<i>F-1 Score</i>	34.1%
Gate kiri	TP	348
	TN	6
	FP	1379
	FN	1965
	Perhitungan Manual	
	<i>Accuracy</i>	63.2%
	<i>Prescision</i>	20.1%
	<i>Recal</i>	98.3%
	<i>F-1 Score</i>	33.4%
flare orange	TP	582
	TN	37
	FP	2329
	FN	775
	Perhitungan Manual	
	<i>Accuracy</i>	36.7%
	<i>Prescision</i>	20%
	<i>Recal</i>	94%
	<i>F-1 Score</i>	32.9%
flare biru	TP	461
	TN	46
	FP	2332
	FN	884
	Perhitungan Manual	
	<i>Accuracy</i>	36.6%
	<i>Prescision</i>	16.5%
	<i>Recal</i>	90.9%
	<i>F-1 Score</i>	28%
flare kuning	TP	656
	TN	26
	FP	2333
	FN	708
	Perhitungan Manual	

Objek	YOLOv8n	
	<i>Accuracy</i>	36.7%
	<i>Prescision</i>	21.9%
	<i>Recal</i>	96.2%
	<i>F-1 Score</i>	35.7%
	TP	166
<i>flare merah</i>	TN	21
	FP	2162
	FN	1374
	Perhitungan Manual	
	<i>Accuracy</i>	41%
	<i>Prescision</i>	7.1%
	<i>Recal</i>	88.8%
	<i>F-1 Score</i>	13.2%

Dari Tabel 4.9 dapat diketahui seluruh parameter pengujian model. Selain itu, dapat dihitung rata-rata akurasi model sebagai berikut.

$$Accuracy = \frac{Accuracy \text{ semua class}}{8}$$

$$Accuracy = \frac{73.9\%+63.1\%+62.5\%+63.2\%+36.7\%+36.3\%+36.7\%+41\%}{8} = 51.6\%$$

$$Prescision = \frac{Prescision \text{ semua class}}{8}$$

$$Prescision = \frac{28,5\%+25,2\%+21,4\%+20,1\%+20\%+16,5\%+21,9\%+7,1\%}{8} = 20,08\%$$

$$Recal = \frac{Recal \text{ semua class}}{8}$$

$$Recal = \frac{95,8\%+83,8\%+83,8\%+98,3\%+94\%+90,9\%+96,2\%+88,8\%}{8} = 91,45\%$$

$$F - 1 \text{ Score} = \frac{F - 1 \text{ Score semua class}}{8}$$

$$F - 1 \text{ Score} = \frac{43,9\%+38,8\%+34,1\%+33,4\%+32,9\%+28\%+35,7\%+13,2\%}{8} = 32,5\%$$

Dengan demikian, hasil *training* model yolov8n memiliki *accuracy* rata-rata sebesar 51.6%; *prescision* 20,08%; *recal* 91,45%; dan *f-1 score* 32,5%. Dari pengujian tersebut sebenarnya model belum layak untuk digunakan karena

keseimbangan jumlah tiap *class* yang tidak seimbang. Dilanjutkan dengan deteksi objek langsung secara *real-time*.

4.6 Deteksi Objek Secara *Real-time*

Deteksi secara langsung dilakukan untuk mengetahui performa model yang sebenarnya atau real. Pengujian ini bertujuan untuk mengetahui bagaimana performa model dalam mendeteksi objek serta menentukan pemetaan lintasan pada lintasan pengujian yang sebenarnya. Pengujian dilakukan pada malam hari dengan menjalankan AUR secara otomatis, AUR akan bermanuver hingga berada didepan objek sehingga dapat membuktikan objek terdeteksi dan akan didapatkan hasil. Proses deteksi objek secara *real-time* dapat dilihat pada dan Gambar 4.33.



Gambar 4. 33 Intensitas Cahaya Pengujian *Real-time*

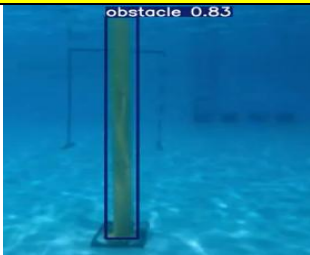
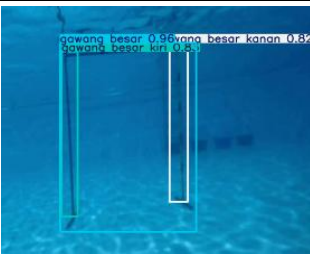
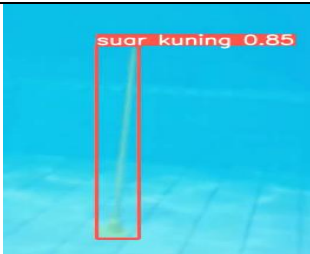

Sesuai Gambar 4.33 dapat diketahui bahwa pengujian dilakukan dengan pencahayaan yang minimum yang dibantu dengan pencahayaan lampu karena dilakukan pada malam hari dan didapatkan intensitas cahaya 102 lux.



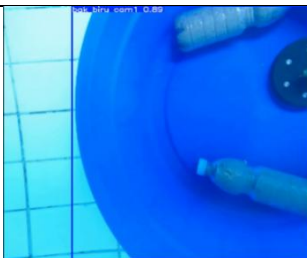
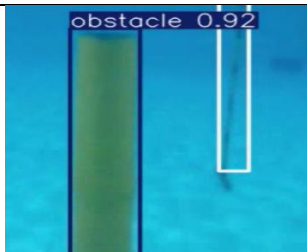

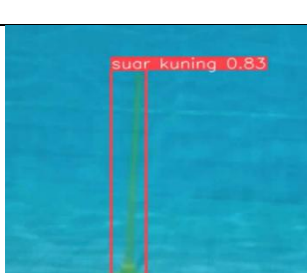


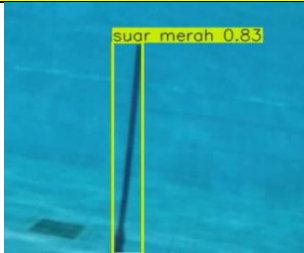



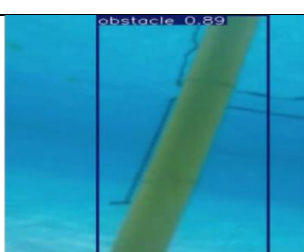
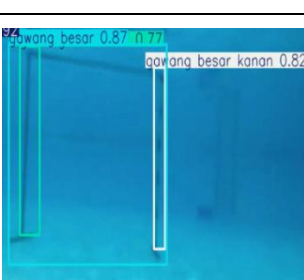
Gambar 4. 34 Proses Uji Deteksi Objek secara langsung






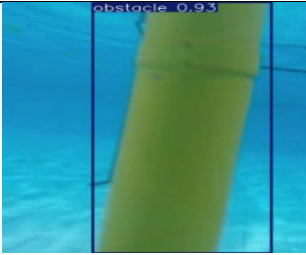
Sesuai Gambar 4.34 dapat diketahui bahwa proses uji deteksi dilakukan pada malam hari dan dibantu dengan cahaya lampu penerangan dengan kondisi air yang sangat jernih. Pengujian dilakukan dilakukan dikolam atlas dengan ukuran 14 x 5,5 x 2,5 m², proses uji coba deteksi dilakukan beberapa kali. Hasil dari deteksi secara *real-time* dapat dilihat pada Tabel 4.10.

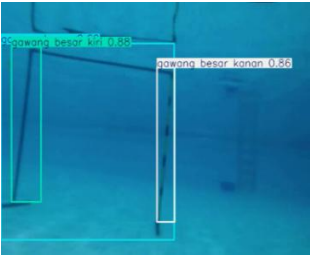

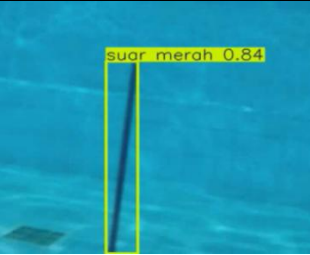



Tabel 4. 10 Hasil Pengujian Deteksi Secara Langsung

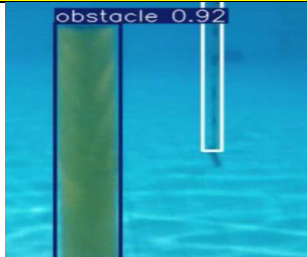
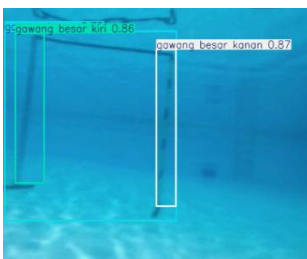
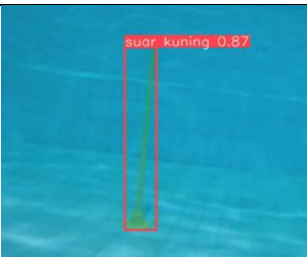


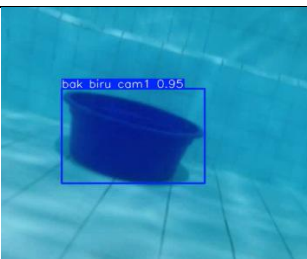
No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
1.	Pertama		<i>Flare orange</i>	83%	17%
2.			<i>Gate utama</i>	96%	4%
			<i>Gate kanan</i>	82%	18%
			<i>Gate kiri</i>	83%	17%
3.			<i>Flare kuning</i>	85%	15%
4.			<i>Flare merah</i>	85%	15%

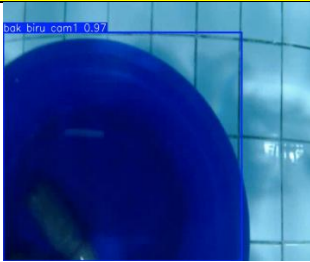

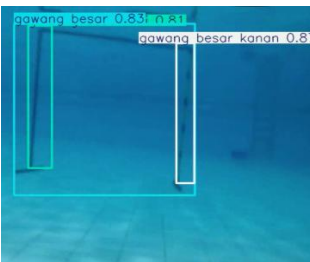

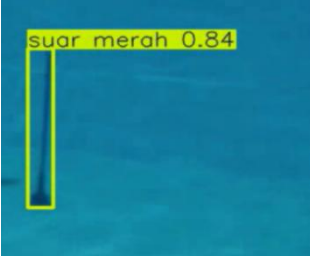
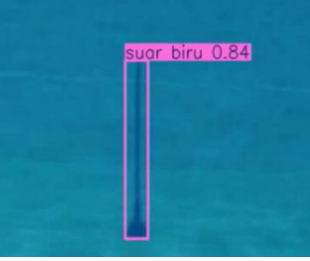
No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
5.			Flare biru	83%	17%
6.			Drum biru (Webcam 1)	96%	4%
7.			Drum biru (Webcam 2)	89%	11%
8.	Kedua		Flare orange	92%	8%
9.			Gate utama	51%	49%
			Gate kanan	86%	14%
			Gate kiri	84%	16%
10.			Flare kuning	83%	17%




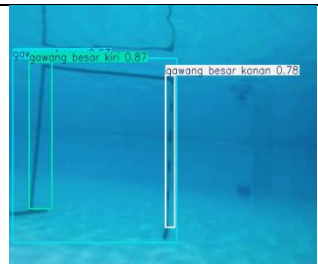

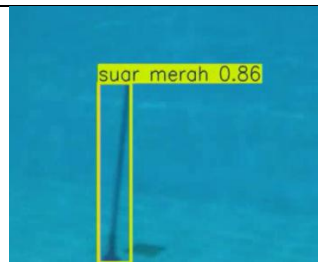
No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
11.			Flare merah	83%	17%
12.			Flare biru	85%	15%
13.			Drum biru (Webcam 1)	96%	4%
14.			Drum biru (Webcam 2)	96%	4%
15.	Ketiga		Flare orange	89%	11%
16.			Gate utama	87%	13%
			Gate kanan	82%	18%
			Gate kiri	77%	23%



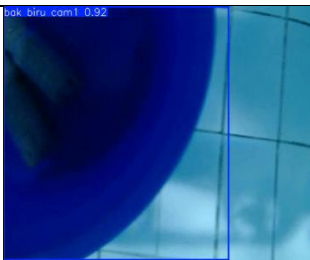
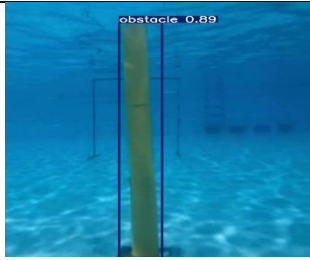
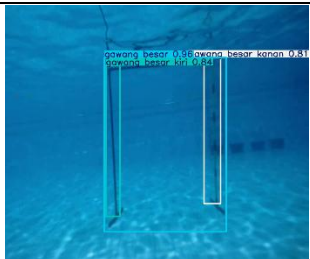

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
17.			Flare kuning	85%	15%
18.			Flare merah	84%	16%
19.			Flare biru	84%	16%
20.			Drum biru (Webcam 1)	96%	4%
21.			Drum biru (Webcam 2)	96%	4%
22.	Keempat		Flare orange	93%	7%





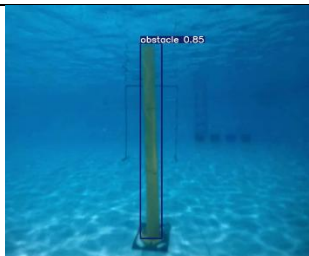
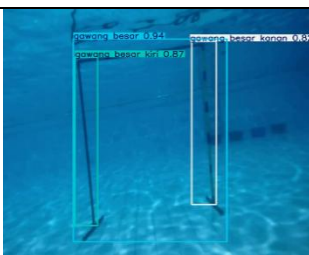
No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
23.			Gate utama	80%	20%
			Gate kanan	86%	14%
			Gate kiri	88%	12%
24.			Flare kuning	83%	17%
25.			Flare merah	84%	16%
26.			Flare biru	87%	13%
27.			Drum biru (Webcam 1)	95%	5%
28.			Drum biru (Webcam 2)	95%	5%

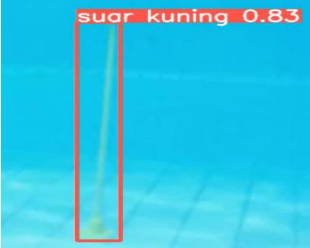



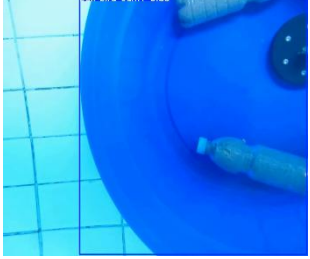
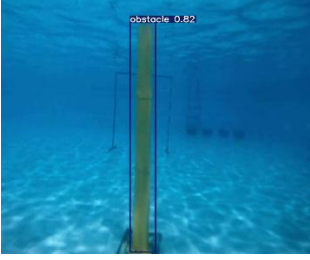
No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
29.	Kelima		Flare orange	92%	8%
30.			Gate utama	75%	25%
			Gate kanan	87%	13%
			Gate kiri	86%	14%
31.			Flare kuning	87%	13%
32.			Flare merah	86%	14%
33.		Flare biru	86%	14%	
34.			Drum biru (Webcam 1)	95%	5%


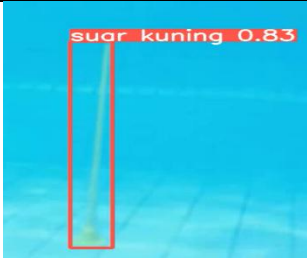



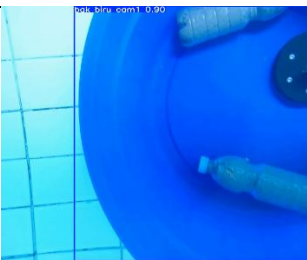
No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
35.			<i>Drum</i> biru (Webcam 2)	97%	3%
36.	Keenam		<i>Flare</i> orange	87%	13%
37.			<i>Gate</i> utama	83%	17%
			<i>Gate</i> kanan	87%	13%
			<i>Gate</i> kiri	81%	19%
38.			<i>Flare</i> kuning	86%	14%
39.			<i>Flare</i> merah	84%	16%
40.			<i>Flare</i> biru	84%	16%

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
41.			<i>Drum</i> biru (Webcam 1)	95%	5%
42.			<i>Drum</i> biru (Webcam 2)	94%	5%
43.	Ketujuh		<i>Flare</i> orange	93%	7%
44.			<i>Gate</i> utama	67%	33%
			<i>Gate</i> kanan	78%	22%
			<i>Gate</i> kiri	87%	13%
45.			<i>Flare</i> kuning	87%	13%
46.			<i>Flare</i> merah	86%	14%

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
47.			Flare biru	83%	17%
48.			Drum biru (Webcam 1)	95%	5%
49.			Drum biru (Webcam 2)	92%	8%
50.	Kedelapan		Flare orange	89%	11%
51.			Gate utama	96%	4%
			Gate kanan	81%	19%
			Gate kiri	84%	16%
52.			Flare kuning	84%	16%

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
53.			Flare merah	84%	16%
54.			Flare biru	83%	17%
55.			Drum biru (Webcam 1)	96%	4%
56.			Drum biru (Webcam 2)	94%	6%
57.	Kesembilan		Flare orange	85%	15%
58.			Gate utama	94%	6%
			Gate kanan	82%	17%
			Gate kiri	87%	13%

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
59.			Flare kuning	87%	13%
60.			Flare merah	85%	15%
61.			Flare biru	83%	17%
62.			Drum biru (Webcam 1)	95%	5%
63.			Drum biru (Webcam 2)	88%	12%
64.	Kesepuluh		Flare orange	82%	18%

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
65.			Gate utama	93%	7%
			Gate kanan	76%	24%
			Gate kiri	83%	17%
66.			Flare kuning	83%	17%
67.			Flare merah	85%	15%
68.			Flare biru	80%	20%
69.			Drum biru (Webcam 1)	95%	5%
70.			Drum biru (Webcam 2)	90%	10%

No.	Percobaan	Gambar Citra	Kelas	Akurasi	Error
	Rata-rata Akurasi dan Error Kelas <i>Flare</i> orange			88,5%	11,5%
	Rata-rata Akurasi dan Error Kelas <i>Gate</i> utama			82,2%	17,8%
	Rata-rata Akurasi dan Error Kelas <i>Gate</i> kanan			82,7%	17,2%
	Rata-rata Akurasi dan Error Kelas <i>Gate</i> kiri			84,0%	16,0%
	Rata-rata Akurasi dan Error Kelas <i>Flare</i> kuning			85,0%	15,0%
	Rata-rata Akurasi dan Error Kelas <i>Flare</i> merah			84,6%	15,4%
	Rata-rata Akurasi dan Error Kelas <i>Flare</i> biru			84,2%	15,8%
	Rata-rata Akurasi dan Error Kelas <i>Drum</i> biru (<i>Webcam</i> 1)			95,4%	4,6%
	Rata-rata Akurasi dan Error Kelas <i>Drum</i> biru (<i>Webcam</i> 2)			93,1%	6,8%
	Total Rata-rata Akurasi dan Error			86,64%	13,36%

Sesuai dari Tabel 4.10 hasil pengujian yang dilakukan sebanyak sepuluh kali menggunakan model YOLOv8n dapat mendeteksi objek pada lintasan dengan rata-rata akurasi 86,64% dan rata-rata nilai error 13,36%.

(Halaman ini sengaja dikosongkan)

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil training dan hasil pengujian identifikasi warna *flare*, *gate*, dan *drum* yang telah dilakukam secara *real-time*, dapat ditarik beberapa kesimpulan pada tugas akhir ini, yaitu:

1. *Autonomous Underwater Robot* (AUR) dapat mengambil *dataset* sesuai dengan jumlah objek pada aturan lomba SAUVC 2025 serta menggunakan konfigurasi *training* sebanyak 5175 *dataset* dan validasi sebanyak 2075 *dataset*. Terdapat 8 *class* pada *dataset* seperti *flare* orange, *gate* utama, *gate* kanan, *gate* kiri, *flare* kuning, *flare* merah, *flare* biru, dan *drum* biru. Proses *training* pada *google collab* membutuhkan waktu 2 jam 27 menit dan didapatkan hasil perhitungan dari *confusion matrix* rata-rata *recall* 91,45%, *accuracy* 51,6%, *prescision* 20,8% dan *f1-score* 32,5%. Hasil tidak seimbang diakibatkan karena kurang samanya jumlah *dataset* tiap *class*.
2. Keseimbangan pada *Autonomous Underwater Robot* (AUR) dapat implementasikan dengan mengintegrasikan HWT905 *Attitude Angle* Sensor. Hasil dari pengujian HWT905 *Attitude Angle* Sensor memiliki tingkat akurasi yang sangat tinggi . Hal ini ditunjukkan dengan nilai rata-rata akurasi per sudut yaitu sudut X sebesar 99,81%; sudut Y sebesar 99,4%; sudut Z sebesar 99,07%. Pengujian ini dilakukan sebanyak sepuluh kali percobaan menggunakan busur derajat dan didapatkan total rata-rata nilai akurasi 99,42% dan total rata-rata nilai error 0,78%.
3. Hasil dari pendeteksian citra lintasan SAUVC dengan metode YOLOv8 memiliki tingkat akurasi yang baik. Hal ini ditunjukkan dengan nilai rata-rata akurasi per *class* yaitu *flare* orange sebesar 88,5% ; *gate* utama sebesar 82,2% ; *gate* kanan sebesar 82,7% ; *gate* kiri sebesar 84% ; *flare* kuning sebesar 85% ; *flare* merah sebesar 84,6% ; *flare* biru sebesar 84,2% ; *drum* biru (*Webcam* 1)

sebesar 95,4%; dan *drum* biru (*Webcam 2*) sebesar 93,1%. Pengujian ini dilakukan sebanyak sepuluh kali percobaan dengan pencahayaan yang rendah dibantu dengan pencahayaan ruangan intensitas 102 lux didapatkan total rata-rata nilai akurasi 86,64% dan total rata-rata nilai error 13,36%.

5.2 Saran

Dari hasil penelitian yang telah dilakukan, terdapat beberapa hal yang perlu diperbaiki dan ditingkatkan pada penelitian selanjutnya untuk mencapai hasil yang lebih maksimal, yaitu:

1. Memperbanyak jumlah *dataset* dan variasi jumlah *dataset*. Contoh pengambilan variasi *dataset* dapat dilakukan di darat sehingga tidak hanya mengambil *dataset* dibawah permukaan air saja. Tujuannya agar nilai akurasi yang dihasilkan dari training menjadi lebih tinggi dan dapat mendeteksi objek lebih baik.
2. Memperhatikan bobot yang dari setiap *class* nya agar sama rata untuk tiap *dataset per class*, sehingga mengurangi nilai perhitungan yang turun drastis.
3. Memperhatikan kualitas cover dari *webcam*, goresan dan retakan pada cover *webcam* dapat mempengaruhi kualitas *dataset* dan hasil *running* menggunakan algoritma YOLOv8. Jika terjadi kerusakan pada cover *webcam* segera dilakukan perbaikan, jika tidak memungkinkan segera diganti dengan *Unit* baru.
4. Memperhatikan kualitas kejernihan air saat pengambilan data dan pengujian data. Kualitas air yang jernih dapat menentukan keberhasilan algoritma YOLOv8 dalam melakukan sistem deteksinya.
5. Memperhatikan kedapatan dalam *enclosure*, jika terdapat kebocoran pada *enclosure* utama dapat dipastikan *enclosure* tersebut akan terjadi embun dan dapat mengganggu deteksi. Diharuskan jika melakukan pengambilan *dataset* dan proses *running* program menggunakan algoritma YOLOv8 tidak terjadi pengembunan.

DAFTAR PUSTAKA

- Danyalson, C., Cahyanti, M., Swedia, E. R., & Sarjono, M. W. (2024). Comparison of Yolov7 and Yolov8 Architectures for Detecting Shirt Collars. *Sebatik*, 28(2). <https://doi.org/10.46984/sebatik.v28i2.2492>
- Dwi Putra, Y., Latifah Husni, N., Silvia Handayani, A., Rasyad, S., Sobri, M., & Negeri Sriwijaya, P. (2022). *Pendeteksi Masker dan Monitoring Suhu menggunakan Webcam dan Sensor Suhu GY-906 Mask Detection and Temperature Monitoring Using Webcam and Temperature Sensor GY-906*. 7(2). <https://doi.org/10.31851/ampere>
- Dwinanto, B., & Yulianto, B. (2024). Rancang Bangun Repeater Lora Rfm95 Dengan Frekuensi 915 Mhz Berbasis Esp32. *Cerdika: Jurnal Ilmiah Indonesia*, 4(2), 109–125. <https://doi.org/10.59141/cerdika.v4i2.752>
- Fang, W., Wang, L., & Ren, P. (2020). Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments. *IEEE Access*, 8, 1935–1944. <https://doi.org/10.1109/ACCESS.2019.2961959>
- Fawwaz Al Maki, W., & Korespondensi, P. (2022). METODE COMPUTATIONAL THINKING UNTUK PENGABDIAN MASYARAKAT DALAM PENINGKATAN KEMAMPUAN BAHASA PEMROGRAMAN PYTHON SISWA SMK (STUDI KASUS: SMK ASSHIDDIQIYAH KARANGPAWITAN, GARUT) COMPUTATIONAL THINKING METHOD FOR THE COMMUNITY SERVICE IN IMPROVING PYTHON PROGRAMMING LANGUAGE CAPABILITIES OF THE VOCATIONAL SCHOOL STUDENTS (CASE STUDY: ASSHIDDIQIYAH VOCATIONAL SCHOOL. *Jurnal Pengabdian Masyarakat Teknologi Informasi dan Informatika (DIMASLOKA)*, 1(1), 1–7.
- Hidayah, N. A., & Rofiqoh, N. (2024). 3383-Article Text-9764-1-10-20241028. 6(EVALUASI SOFTWARE VISUAL STUDIO CODE MENGGUNAKAN METODE QUETIONNAIRES NELSEN’S ATTRIBUTES OF USABILITY), 1–10. <https://doi.org/https://doi.org/10.32520/jupel.v6i3.3383>
- Indah Kusumawati, W., Pratikno, H., & Pradeska Admaja, Y. (2021). Sistem Penghitung Jumlah Pengunjung Restoran Menggunakan Kamera Berbasis

- Single Shot Detector (SSD). *Journal of Technology and Informatics (JoTI)*, 3(1), 19–26. <https://doi.org/10.37802/joti.v3i1.197>
- Maulana, A., Auliatunnajah, F., Rosidin, N., Ramadien Rizki Darmawan, M., & Rosyani, P. (2024). Implementasi OpenCV dengan Metode Image Thresholding pada Gambar. *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, 2(1). <https://jurnalmahasiswa.com/index.php/aidanspk>
- Muzammil, M. A. A., & Indraswari, R. (2024). Pengembangan Arsitektur Model YOLOv8 untuk Meningkatkan Performa Object Detection pada Varian Boks Warehouse Palletizing. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 6(2), 19–30. <https://doi.org/10.28926/ilkomnika.v6i2.642>
- Nurhadi, H. (2019). *Estimasi Lintasan AUV 3 Dimensi (3D) Dengan Ensemble Kalman Filter !*
- Oni, M., Kanata, B., & Ratnasari, D. (2021). *MENENTUKAN LUAS OBJEK CITRA DENGAN TEKNIK SEGMENTASI BERDASARKAN WARNA PADA RUANG WARNA HSV Determining the Image Object Area Using Color-Based Segmentation Technique in HSV Color Space* (Vol. 8, Nomor 2).
- Pradiftha, A., & Elektro, J. T. (2019). *Identifikasi Gas terlarut Minyak Transformator dengan Menggunakan Logika Fuzzy Menggunakan Metode TDCG untuk Menentukan Kondisi Transformator 150 KV*. 1(1), 11–15.
- Prakosa, S., Nugraha, A., Sunuharjo, L., & Atiq, M. '. (2024). *Komunikasi Arduino I2C, SPI dan UART*. 4, 80–85. <https://doi.org/10.62951/switch.v2i4.18>
- Rabbani, H. A., Rahman, M. A., & Rahayudi, B. (2021). *Perbandingan Ruang Warna RGB dan HSV dalam Klasifikasi Kematangan Biji Kopi* (Vol. 5, Nomor 6). <http://j-ptiik.ub.ac.id>
- Rahardi, M.R., Adam, K.B. and Raharjo, J. (2022) 'Analisis Keekonomian Sederhana Baterai Lithium Ion (Li-ion)', *eProceedings of Engineering*, 9(5), pp. 1998–2007.
- Razak, A. A., Muttaqin, A., Pradipta, E., & Elektro, J. T. (2021). Analisis Pembatasan Thread Cuda GPU Computing pada Image Denoising. Dalam *Jurnal EECCIS* (Vol. 15, Nomor 1). <https://jurnaleeccis.ub.ac.id/>

- Romadloni, F., Endrasmono, J., Maulana, Z., Putra, A., Khumaidi, A., Rachman, I., Adhitya, R. Y., Studi, P., Otomasi, T., Tenkik, J., Kapal, K., Perkapalan, P., & Surabaya, N. (t.t.). *Identifikasi Warna Buoy Menggunakan Metode You Only Look Once Pada Unmanned Surface Vehicle*. <https://journal.trunojoyo.ac.id/triac>
- Rusdi, M. (2020). *SIMULASI DAN ANALISIS KINERJA BUCK CONVERTER DENGAN ZERO-VOLTAGE SWITCHING (ZVS)*. 9(1).
- Wicaksono, D., Almeyda, P., Mikola, I., Putra, M., & Malihatuningrum, L. (2024). *Analisis Perbandingan Metode Pra Pemrosesan Citra untuk Deteksi Tepi Canny pada Citra Berbagai Kondisi Jalan menggunakan Bahasa Pemrograman Python*. JUTIKOMP.
- Yurkha Kusuma, A., Pratikno, H., Puspasari, I., Studi, P., & Teknik Komputer, J. (2020). RANCANG BANGUN ALAT PELIPAT BAJU OTOMATIS MENGGUNAKAN ARDUINO UNO. Dalam *Ira Puspasari JCONES* (Vol. 9, Nomor 2). <http://jurnal.stikom.edu/index.php/jcone>

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1 Sertifikat Juara 2 Internasional SAUVC



Lampiran 1. 1 Sertifikat Juara 2 Internasional SAUVC

(Halaman ini sengaja dikosongkan)

Lampiran 2 Biodata Mahasiswa

Nama : Erutan Dwi Sajiwo
NRP : 0921040008
Program Studi : D4 Teknik Otomasi
Agama : Islam
Status : Belum Menikah
Alamat Asal : Baron, Boro RT. 04/ RW. 02, Kedungwaru,
Tulungagung
Nomor Telepon : 082332529611
Jenis Kelamin : Laki-Laki
Email : derutandwisajiwo@gmail.com
Tempat, Tanggal Lahir : Tulungagung, 04 Juli 2001
Nama Orang Tua/ Wali : Muri
Alamat Orang Tua / Wali : Baron, Boro RT. 04/ RW. 02, Kedungwaru,
Tulungagung
Telepon Orang Tua / Wali : 085236906433
Riwayat Pendidikan



PENDIDIKAN FORMAL			
Pendidikan	Tahun	Tempat Pendidikan	Jurusan
Diploma 4	2021- Sekarang	Politeknik Perkapalan Negeri Surabaya	Teknik Otomasi
SMA	2017-2021	SMKN 3 Boyolangu	Teknik Otomasi Industri
SMP	2014-2017	SMPN 1 Karangrejo	-
SD	2008-2014	SDI AL-BADAR	-