

**DETEKSI DAN KLASIFIKASI SAMPAH PESISIR PANTAI BUNGA  
KABUPATEN BATUBARA MENGGUNAKAN ALGORITMA  
*YOU ONLY LOOK ONCE V8 DAN RESIDUAL NETWORK-50***

**SKRIPSI**

**YOGI PIRANDA LIMBONG  
201401121**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**DETEKSI DAN KLASIFIKASI SAMPAH PESISIR PANTAI BUNGA  
KABUPATEN BATUBARA MENGGUNAKAN ALGORITMA  
*YOU ONLY LOOK ONCE V8 DAN RESIDUAL NETWORK-50***

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Ilmu Komputer**

**YOGI PIRANDA LIMBONG**

**201401121**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

## PERSETUJUAN

Judul : DETEKSI DAN KLASIFIKASI SAMPAH PESISIR  
 PANTAI BUNGA KABUPATEN BATUBARA  
 MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE V8*DAN *RESIDUAL NETWORK-50*

Kategori : SKRIPSI

Nama : YOGI PIRANDA LIMBONG

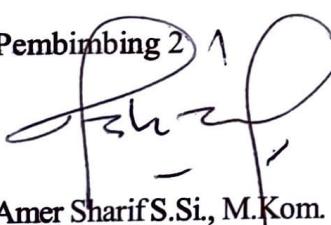
Nomor Induk Mahasiswa : 201401121

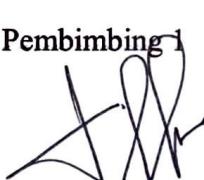
Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
 UNIVERSITAS SUMATERA UTARA

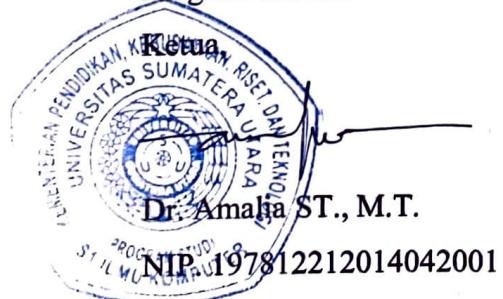
Medan, 27 Mei 2024

Komisi Pembimbing :

Pembimbing 2  
  
 Amer Sharif S.Si., M.Kom.  
 NIP. 196910212021011001

Pembimbing 1  
  
 Dr. T. Henny Febriana Harumy S.Kom., M.Kom.  
 NIP. 198802192019032016

Diketahui/Disetujui Oleh  
 Program Studi S-1 Ilmu Komputer



## **PERNYATAAN**

**DETEKSI DAN KLASIFIKASI SAMPAH PESISIR PANTAI BUNGA  
KABUPATEN BATUBARA MENGGUNAKAN ALGORITMA  
*YOU ONLY LOOK ONCE V8 DAN RESIDUAL NETWORK-50***

## **SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 27 Mei 2024



Yogi Piranda Limbong

201401121

## PENGHARGAAN

*Bismillahirrahmanirrahim*, segala puji dan syukur penulis panjatkan kepada Allah *Subhanahu Wa Ta’ala* atas seluruh limpahan rahmat dan hidayah-Nya sehingga penulis dapat diberi kesehatan dalam penyusunan skripsi ini sebagai syarat untuk mendapatkan gelar Sarjana Komputer di Program Studi S-1 Ilmu Komputer, Universitas Sumatera Utara. Shalawat dan salam tidak lupa diucapkan kepada Rasulullah *Shalallaahu ‘Alayhi Wasallam*, karena baginda rasulullah telah menjadi perantara kita untuk dapat memanfaatkan ilmu yang diberikan Allah SWT.

Dengan rasa penuh hormat, pada kesempatan ini penulis mengucapkan terima kasih kepada Ibu Maiharti atas segala bentuk perjuangan, kasih sayang, dan perlindungan dengan doa-doa yang dipanjatkan untuk penulis. Dan terima kasih kepada Bapak Ridho Limbong atas dukungan dan kasih sayang yang selalu ada di setiap langkah penulis. Terima kasih atas setiap dukungan yang diberikan hingga penulis dapat berada di titik ini.

Penyusunan skripsi ini tidak terlepas atas bantuan, dukungan, dan bimbingan oleh banyak pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi S.Kom., M.Kom selaku Sekretaris Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Universitas Sumatera Utara.
5. Ibu Dr. T. Henny Febriana Harumy S.Kom., M.Kom. sebagai Dosen Pembimbing I yang membantu penulis dalam menyelesaikan skripsi dengan memberikan bimbingan, kritik, motivasi, dan saran.
6. Bapak Amer Sharif S.Si., M.Kom. selaku Dosen Pembimbing II yang telah memberi banyak bimbingan, masukan, dan bantuan yang berharga kepada penulis selama penyusunan skripsi ini.

7. Bapak Handrizal S.Si., M.Comp.Sc. selaku Dosen Pembimbing Akademik yang telah memberi banyak dukungan dan motivasi kepada penulis.
8. Seluruh bapak dan ibu dosen Fasilkom-TI Universitas Sumatera Utara, khususnya dosen Program Studi S-1 Ilmu Komputer yang telah mendidik dan memberi wawasan serta moral yang berharga, baik di bangku perkuliahan maupun setelah lulus.
9. Seluruh Staf Pegawai Fasilkom-TI Universitas Sumatera Utara yang dengan baik hati memberikan bantuan kepada penulis selama periode perkuliahan sampai kepada tahap penyusunan skripsi ini.
10. Bapak Ridho Limbong dan Ibu Maiharti selaku orang tua yang tidak pernah hentinya memberikan motivasi, dukungan, doa serta saran dan kasih sayang penuh kepada penulis dalam menuntaskan pendidikan.
11. Kakak kandung tercinta Yuni Sartika Limbong dan juga Adik kandung tercinta Tiara Sabilia Limbong yang selalu mendukung serta mendoakan penulis dalam menjalankan aktivitas kuliah hingga akhirnya menyelesaikan tugas akhir.
12. Kepada sc awak khususnya Tjahyo Edhi Santoso yang telah mendoakan dan membantu dalam proses skripsi ini, juga memberikan spirit agar lebih semangat dalam menyelesaikan skripsi ini.
13. Kepada bude dan abang sepupu yang selalu mendoakan, mendukung, dan memberi saran serta motivasi untuk penulis.
14. Sahabat Tentrem Squad yaitu bang Jimmi Eduard Simangunsong dan bang Irfan Akbari Habibi yang selalu sabar dalam membantu menulis dan memberi saran, dukungan, serta motivasi kepada penulis.
15. Sahabat penulis Andrew, TM Rezha, Farel, Doli, Alfan, Chindy, Susi, Sukiya, Salsabila, Wina, Difanie, Zahra yang mendukung juga memberikan masukan positif.
16. Sahabat perkuliahan Kom C stambuk 2020 yang ramai, meriah, dan akan selalu terkenang, terima kasih untuk seluruh pembelajaran barunya, baik dari aspek akademis maupun bekal dalam menjalani kehidupan dunia dan akhirat.
17. Sahabat traveling Yudha, Torik, Dio, Dimas, Reza, Odi, Nanda, Izal, Abdul, Enzo, Adit yang terus memberi dukungan serta memberi keceriaan untuk penulis.

Dan seluruh pihak yang telah memberi dukungan serta doa baik semoga Allah *Subhanahu Wa Ta ‘ala* senantiasa melimpahkan keberkahan serta kebaikan dari semua dukungan yang telah diberikan kepada penulis dan hasil penelitian ini dapat memberi manfaat maupun inspirasi untuk kedepannya.

Medan, 27 Mei 2024

Penulis,



Yogi Piranda Limbong

## ABSTRAK

Terdapat 3 golongan jenis sampah, di antaranya sampah organik yaitu sampah yang berasal dari sisa makhluk hidup dan mudah mengalami proses pembusukan alami. Sampah anorganik adalah sampah yang sulit terurai atau mengalami proses pembusukan yang memerlukan waktu sangat lama untuk mengalami dekomposisi, terakhir sampah bahan berbahaya dan beracun (B3) yaitu jenis sampah yang mengandung bahan kimia berpotensi berbahaya serta zat-zat beracun. Memiliki komposisi kimia yang dapat membahayakan kesehatan manusia dan lingkungan, baik secara langsung maupun melalui paparan jangka panjang. Dampak dari sampah di pesisir tidak terbatas pada sekadar sektor ekonomi dan pariwisata, juga mencakup terhadap kehidupan organisme dalam ekosistem laut, pencemaran atau kerusakan lingkungan hidup, serta ekosistem perairan dan dapat menimbulkan risiko terhadap kesehatan manusia. Untuk itu dibutuhkan sistem yang dapat mendeteksi sampah dan mengklasifikasikan jenis sampah tersebut. Dalam prosesnya, memasukkan gambar sampah ke sistem untuk dilakukan proses deteksi menggunakan algoritma YOLOv8 melalui *bounding box* dan hasilnya deteksi akan kembali di klasifikasikan menggunakan metode ResNet-50 untuk menggolongkan sampah ke dalam 3 jenis, yaitu organik, anorganik, dan bahan berbahaya beracun (B3). Sampah yang dapat dideteksi adalah sampah daun, kayu, sampah buah, plastik, kertas atau tisu, *styrofoam*, dan rokok. Hasil pengujian model YOLO mendapat nilai *accuracy* 0.87, *precision* 0.90, *recall* 0.89 dan *F1 Score* 0.89 dan hasil pengujian model ResNet-50 mendapat *accuracy* 0.89, *precision* 0.89, *recall* 0.89 dan *F1 Score* 0.89, serta hasil pengujian sistem secara keseluruhan mendapat 85% dari 20 kali percobaan.

**Kata Kunci:** Sampah Organik, Anorganik, Bahan Berbahaya dan Beracun (B3), You Only Look Once, Residual Network 50.

**DETECTION AND CLASSIFICATION OF COASTAL GARBAGE ON BUNGA  
BEACH BATUBARA DISTRICT USING THE YOU ONLY LOOK ONCE  
V8 ALGORITHM AND RESIDUAL NETWORK-50**

**ABSTRACT**

There are 3 categories of waste, namely organic waste which originates from living organisms and easily undergoes natural decomposition processes. Inorganic waste is waste that is difficult to degrade or undergoes a very long decomposition process, and hazardous and toxic waste (B3) which contains potentially dangerous chemicals and toxic substances. It has chemical compositions that can endanger human and environmental health, both directly and through long-term exposure. The impact of waste on the coast is not limited to just the economic and tourism sectors but also extends to the lives of organisms in marine ecosystems, environmental pollution or damage, as well as aquatic ecosystems, and can pose risks to human health. Therefore, a system is needed that can detect and classify such waste. In the process, images of waste are inputted into the system for detection using the YOLOv8 algorithm via bounding boxes, and the detected results are then classified using the ResNet-50 method to categorize the waste into 3 types: organic, inorganic, and hazardous and toxic waste (B3). Detectable waste includes leaf waste, wood waste, fruit waste, plastic, paper or tissue, styrofoam, and cigarette waste. The testing results of the YOLO model obtained an accuracy value of 0.87, precision of 0.90, recall of 0.89, and F1 Score of 0.89, and the testing results of the ResNet-50 model obtained an accuracy of 0.89, precision of 0.89, recall of 0.89, and F1 Score of 0.89, with the overall system testing results achieving 85% out of 20 attempts.

***Keywords:*** *Organic trash, inorganic, hazardous and toxic materials, You Only Look Once, Residual Network 50.*

## DAFTAR ISI

<b>PERSETUJUAN .....</b>	ii
<b>PERNYATAAN .....</b>	iii
<b>PENGHARGAAN .....</b>	iv
<b>ABSTRAK .....</b>	vii
<b>ABSTRACT .....</b>	viii
<b>DAFTAR ISI .....</b>	ix
<b>DAFTAR TABEL .....</b>	xii
<b>DAFTAR GAMBAR .....</b>	xiii
<b>BAB 1 PENDAHULUAN.....</b>	1
<b>1.1 Latar Belakang .....</b>	1
<b>1.2 Rumusan Masalah.....</b>	4
<b>1.3 Batasan Masalah .....</b>	4
<b>1.4 Tujuan Penelitian.....</b>	5
<b>1.5 Manfaat Penelitian.....</b>	5
<b>1.6 Metodologi Penelitian .....</b>	5
<b>1.7 Penelitian Relevan.....</b>	7
<b>1.8 Sistematika Penulisan .....</b>	9
<b>BAB 2 LANDASAN TEORI.....</b>	10
<b>2.1 Deep Learning .....</b>	10
<b>2.2 Object Detection.....</b>	11
<b>2.3 You Only Look Once (YOLOv8).....</b>	11
<b>2.3.1 Bounding box.....</b>	14
<b>2.4 Residual Network 50 .....</b>	14
<b>2.5 Sampah.....</b>	15
<b>2.5.1 Organik .....</b>	16

2.5.2	<i>Anorganik</i> .....	16
2.5.3	<i>Bahan Berbahaya dan Beracun (B3)</i> .....	17
<b>2.6</b>	<b>Sumber Daya Kelautan Berkelanjutan</b> .....	17
<b>BAB 3 ANALISIS DAN PERANCANGAN</b> .....		18
<b>3.1</b>	<b>Analisis</b> .....	18
3.1.1	<i>Analisis masalah</i> .....	18
3.1.2	<i>Analisis data</i> .....	18
3.1.3	<i>Analisis kebutuhan</i> .....	19
<b>3.2</b>	<b>Arsitektur Umum</b> .....	20
<b>3.3</b>	<b>Pemodelan Sistem</b> .....	21
3.3.1	<i>Use case diagram</i> .....	21
3.3.2	<i>Activity diagram</i> .....	22
3.3.3	<i>Sequence diagram</i> .....	24
3.3.4	<i>Flowchart sistem</i> .....	24
<b>3.4</b>	<b>Perancangan Interface</b> .....	26
3.4.1	<i>Halaman utama website</i> .....	26
3.4.2	<i>Halaman deteksi dengan YOLO</i> .....	27
3.4.3	<i>Halaman klasifikasi dengan ResNet-50</i> .....	27
<b>BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM</b> .....		29
<b>4.1</b>	<b>Implementasi Sistem</b> .....	29
4.1.1	<i>Landing page</i> .....	29
4.1.2	<i>Halaman upload foto YOLO</i> .....	29
4.1.3	<i>Hasil Deteksi You Only Look Once</i> .....	30
4.1.4	<i>Proses Klasifikasi ResNet-50</i> .....	30
<b>4.2</b>	<b>Pengujian Sistem</b> .....	31
4.2.1	<i>Program algoritma You Only Look Once</i> .....	32
4.2.2	<i>Pengujian model YOLO</i> .....	37

4.2.3	<i>Program metode ResNet-50</i> .....	40
4.2.4	<i>Pengujian model dari ResNet-50</i> .....	52
4.3	<b>Hasil Pengujian Sistem</b> .....	54
4.4	<b>Faktor Pengaruh Hasil Prediksi</b> .....	56
<b>BAB 5 PENUHUP</b>	.....	57
5.1	<b>Kesimpulan</b> .....	57
5.2	<b>Saran</b> .....	58
<b>DAFTAR PUSTAKA</b>	.....	59

**DAFTAR TABEL**

<b>Tabel 4.1</b> Daftar Kelas dan Jenis Sampah .....	31
<b>Tabel 4.2</b> Pengujian Model YOLO .....	37
<b>Tabel 4.3</b> Tabel Classification Report 1 .....	50
<b>Tabel 4.4</b> Tabel Classification Report 2 .....	50
<b>Tabel 4.5</b> Pengujian Model ResNet-50 .....	52
<b>Tabel 4.6</b> Tabel Pengujian Sistem .....	54

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Ilustrasi Deep Learning .....	10
<b>Gambar 2.2</b> YOLO Real Time Object Detection.....	13
<b>Gambar 2.3</b> Arsitektur You Only Look Once.....	14
<b>Gambar 2.4</b> Prediksi Lokasi Bounding Box .....	14
<b>Gambar 2.5</b> Arsitektur ResNet-50 .....	15
<b>Gambar 2.6</b> Sampah Organik.....	16
<b>Gambar 2.7</b> Sampah Anorganik.....	16
<b>Gambar 2.8</b> Sampah B3.....	17
<b>Gambar 3.1</b> Arsitektur Umum .....	20
<b>Gambar 3.2</b> Use Case Diagram.....	21
<b>Gambar 3.3</b> Activity Diagram.....	23
<b>Gambar 3.4</b> Sequence Diagram .....	24
<b>Gambar 3.5</b> Flowchart Sistem .....	25
<b>Gambar 3.6</b> Halaman Utama .....	26
<b>Gambar 3.7</b> Halaman Deteksi YOLO .....	27
<b>Gambar 3.8</b> Halaman Klasifikasi ResNet-50.....	27
<b>Gambar 4.1</b> Halaman Utama .....	29
<b>Gambar 4.2</b> Halaman Deteksi YOLO .....	30
<b>Gambar 4.3</b> Halaman Hasil Deteksi.....	30
<b>Gambar 4.4</b> Halaman Klasifikasi ResNet-50.....	31
<b>Gambar 4.5</b> Labeling dan Augmentasi di Roboflow.....	32
<b>Gambar 4.6</b> Library .....	32
<b>Gambar 4.7</b> Install Ultralytics.....	33
<b>Gambar 4.8</b> Import Ultralytics.....	33
<b>Gambar 4.9</b> Import YOLO .....	33
<b>Gambar 4.10</b> Model YOLOv8 .....	33
<b>Gambar 4.11</b> Pelatihan Model .....	34
<b>Gambar 4.12</b> Progres Pelatihan .....	34
<b>Gambar 4.13</b> Import Locale .....	34

<b>Gambar 4.14</b> Path gambar yang dideteksi dan model ‘best.pt’ .....	35
<b>Gambar 4.15</b> Hasil Deteksi YOLO .....	35
<b>Gambar 4.16</b> Grafik Pelatihan Model .....	36
<b>Gambar 4.17</b> Library .....	40
<b>Gambar 4.18</b> Model ResNet-50 .....	41
<b>Gambar 4.19</b> Membekukan Lapisan Model .....	41
<b>Gambar 4.20</b> Menghapus Lapisan Terakhir .....	42
<b>Gambar 4.21</b> Menambahkan Lapisan Kustom .....	42
<b>Gambar 4.22</b> Mengatur ‘resnet_model.trainable’ .....	42
<b>Gambar 4.23</b> Kompilasi Model.....	43
<b>Gambar 4.24</b> Menghubungkan ke Google Drive.....	43
<b>Gambar 4.25</b> Mendefenisikan Generator .....	43
<b>Gambar 4.26</b> Eksplorasi dan Analisis Dataset.....	44
<b>Gambar 4.27</b> Grafik Jumlah Data per Kelas.....	44
<b>Gambar 4.28</b> Augmentasi (1).....	45
<b>Gambar 4.29</b> Augmentasi (2).....	45
<b>Gambar 4.30</b> Pelatihan Dataset.....	46
<b>Gambar 4.31</b> Hasil Pelatihan .....	47
<b>Gambar 4.32</b> Membuat Grafik Training dan Validation.....	47
<b>Gambar 4.33</b> Grafik Accuracy.....	48
<b>Gambar 4.34</b> Grafik Loss .....	48
<b>Gambar 4.35</b> Evaluasi dengan Metrik .....	49
<b>Gambar 4.36</b> Nilai Accuracy Model .....	49
<b>Gambar 4.37</b> Proses Prediksi Gambar (1) .....	51
<b>Gambar 4.38</b> Proses Prediksi Gambar (2) .....	51
<b>Gambar 4.39</b> Proses Prediksi Gambar (3) .....	51
<b>Gambar 4.40</b> Hasil Prediksi.....	52

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Menurut (Septiani & Susilawati, 2023) mengatakan bahwa indonesia adalah negara maritim, sebanyak 16,42 juta jiwa penduduk indonesia yang tinggal di wilayah pesisir. Wilayah pesisir merupakan tempat bertemunya daratan dan lautan, daerah darat pesisir termasuk bagian daratan baik kering maupun terendam air dan terus terpengaruh fitur laut seperti pasang surut, angin laut dan intrusi air asin. Wilayah pesisir ini merupakan tempat di mana ekosistem daratan dan laut saling berinteraksi secara dinamis, membentuk sebuah kawasan yang sangat dipengaruhi oleh aktivitas manusia. Di sini, terdapat intensitas kegiatan manusia yang tinggi, seperti pusat pemerintahan, pemukiman, industri, pelabuhan, budidaya perikanan, pertanian, dan pariwisata, yang semuanya memiliki dampak yang signifikan terhadap ekosistem tersebut (Husain & Saleh, 2022).

Keputusan Presiden Nomor 16 Tahun 2017 kebijakan maritim indonesia menekankan pada perlindungan lingkungan pesisir dan laut sampah, khususnya sampah plastik. Hal ini dikarenakan sampah plastik membutuhkan waktu yang sangat lama untuk terurai sehingga mengganggu ekosistem laut (Cahyadi et al., 2021). Menurut (Harefa et al., 2022) terdapat jenis sampah yang mencemari pantai, yaitu sampah dari kegiatan wisata dan sampah laut. Pengelolaan sampah dapat menjadi ancaman serius bagi kelangsungan dan kelestarian kawasan wisata alam.

Dampak dari sampah laut tidak hanya mempengaruhi sektor ekonomi dan pariwisata, tetapi juga menimbulkan gangguan terhadap kehidupan organisme dalam ekosistem laut, serta menyebabkan masalah terkait hama, bau yang tidak sedap, dan ancaman terhadap kesehatan manusia. Sampah laut juga menjadi penyebab utama pencemaran dan kerusakan lingkungan hidup serta ekosistem perairan, menciptakan tantangan serius yang memerlukan perhatian dalam upaya pencegahan dan penanganannya (Ayunda et al., 2023). Dengan adanya kesadaran masyarakat terhadap dampak bahaya sampah, dapat mempengaruhi sumber daya

kelautan yang lebih baik. Pencemaran yang terjadi di pesisir pantai menyebabkan kerusakan ekosistem laut, mencakup sampah plastik dan logam berat.

Plastik dapat terurai menjadi sampah berukuran mikro yang berbahaya bagi organisme laut dan menimbulkan resiko terhadap fungsi fisiologis tubuh seperti gangguan pertumbuhan, disfungsi organ, dan gangguan laju reproduksi biota laut disebabkan oleh zat *mikroplastic* serta akumulasi zat *hydrophobic* berbahaya dari perairan sekitar sebelum dapat tertelan oleh hewan laut (Herdiansyah et al., 2021). Begitu pula dengan peningkatan logam berat di perairan laut yang berdampak buruk terhadap ekosistem karena memiliki tingkat toksitas yang tinggi, bersifat persisten, dan dapat menyebabkan bioakumulasi dan biomagnifikasi pada rantai makanan. Pencemaran daerah pantai juga dapat berdampak buruk bagi kelangsungan sumber daya kelautan, contohnya seperti kurangnya minat untuk berwisata ke area pantai.

Dalam rangka mengatasi permasalahan sampah pesisir, pemanfaatan teknologi kecerdasan buatan, khususnya pada bidang *neural network* adalah hal yang tepat. Penggunaan algoritma YOLO sudah terbukti efektif dalam dalam kasus deteksi objek. Deteksi objek merupakan cabang *computer vision* yang bertujuan untuk mengenali, memprediksi objek, dan membuat *bounding box* (Junzhe et al., 2023). Dengan pemanfaatan teknologi kecerdasan buatan, informasi terkait suatu objek dapat diekstraksi untuk keperluan yang beragam. Salah satu metode yang digunakan adalah deteksi objek, yang bertujuan untuk mengidentifikasi lokasi objek tertentu dalam suatu gambar. Proses ini melibatkan penciptaan kotak pembatas atau *bounding box* yang mengelilingi objek yang terdeteksi (Nufus, N et.al, 2021).

Algoritma YOLO merupakan salah satu pendekripsi objek yang memiliki akurasi yang baik dan mampu mendekripsi objek secara *real-time* (Amwin, 2021). Algoritma YOLO melakukannya tugasnya dengan membagi *inputan* gambar menjadi beberapa kotak, yang masing-masing berisi sejumlah objek tertentu pada *bounding box*. Dengan mendekripsi setiap *bounding box*, algoritma memperoleh informasi setiap kategori, serta posisi dan ukurannya *bounding box*, sehingga menghasilkan deteksi yang cepat pada proses prediksi. Cara kerja YOLO melibatkan penerapan satu jaringan syaraf yang memproses keseluruhan gambar secara serentak. Proses ini dimulai dengan membagi gambar menjadi wilayah-

wilayah yang kemudian dianalisis untuk memprediksi kotak pembatas dan probabilitas keberadaan objek di dalamnya. Setiap kotak wilayah pembatas dievaluasi probabilitasnya untuk menentukan apakah mengandung objek atau tidak. Kemudian, kotak pembatas dengan nilai probabilitas tertinggi dipilih sebagai pemisah antar objek. YOLO dapat mengklasifikasikan objek hingga 155 *frame per second* (FPS) dalam waktu *real* sekaligus mencapai dua kali lipat presisi rata-rata (mAP) dari pengklasifikasi objek lainnya (Wahyutama & Hwang, 2022).

Metode klasifikasi berikutnya yaitu ResNet-50 yang merupakan singkatan dari *Residual Network*. ResNet-50 adalah jaringan saraf konvolusional dengan kedalaman 50 layer. ResNet-50 memiliki performa lebih baik dari metode lain dengan memiliki 48 lapisan konvolusi bersama dengan 1 *MaxPool* dan 1 lapisan *Pooling* Rata-rata. Ini memiliki operasi *floating-point*  $3,8 \times 10^9$ . Dalam mencegah terjadinya masalah kehilang *gradien* (*vanishing gradient problem*), ResNet-50 memperkenalkan *identity shortcut connection* atau yang disebut dengan *residual block* di mana dapat melewati 1 lapisan atau layer (Reynaldi Tanjung & Juwianto, 2022). *Residual block* terdiri dari dua atau tiga *sequential convolutional layers* yang dalam satu *residual block* memiliki *size* 1x1, 3x3 dan 1x1 dan *shortcut layer* atau *skip connection* yang menghubungkan *input* dari lapisan pertama dan *outputnya* yang melewati beberapa lapisan model.

Mengintegrasikan YOLOv8 dan ResNet-50 dalam deteksi dan klasifikasi sampah pesisir pantai di Kabupaten Batubara memberikan sejumlah keuntungan signifikan. Pertama, dengan memanfaatkan kemampuan ResNet-50 dalam mengekstraksi fitur yang mendalam dan beragam dari gambar, sistem dapat memperoleh representasi fitur yang lebih kaya, meningkatkan kualitas klasifikasi sampah. Kedua, dengan kombinasi YOLOv8 yang cepat dan akurat dalam mendeteksi objek serta ResNet-50 yang mendalam dalam pemahaman fitur gambar, kesalahan klasifikasi dapat dikurangi dan identifikasi jenis sampah menjadi lebih tepat. Terakhir, kekuatan komplementer dari kedua algoritma ini memungkinkan penggabungan yang efisien untuk meningkatkan kinerja sistem secara keseluruhan.

Dengan demikian, integrasi ini membentuk sistem yang lebih kuat dan andal dalam mengatasi tantangan deteksi dan klasifikasi sampah pesisir pantai, mengoptimalkan potensi masing-masing algoritma untuk mencapai tujuan akhir secara efektif.

Penelitian ini akan mendeteksi dan mengklasifikasikan sampah pesisir pantai menjadi sampah organik, anorganik, dan bahan berbahaya beracun (B3). Cara kerjanya yaitu dengan memasukkan satu gambar yang terdapat beberapa objek sampah, dengan menggunakan YOLOv8 sebagai pendekripsi objek dan ResNet-50 untuk klasifikasi sampah yang terdeteksi, sehingga dapat membantu membedakan sampah ke dalam jenis organik, anorganik dan sampah B3. Selain itu peneliti juga menggunakan *dataset* sendiri dan diharapkan mendapat tingkat akurasi yang tinggi dalam penelitiannya. *Dataset* yang diambil yaitu berupa gambar sampah yang berada di pinggiran area pantai yang tergolong ke dalam kelas yang ditentukan.

## 1.2 Rumusan Masalah

Kurangnya kesadaran masyarakat terhadap dampak bahaya masalah sampah yang cukup serius, dapat berakibat fatal bagi berbagai sektor, seperti ekonomi dan pariwisata. Dampak ini dapat menimbulkan gangguan signifikan terhadap kehidupan ekosistem laut, menimbulkan hama dan bau tidak sedap, serta menjadi ancaman serius bagi kesehatan manusia. Oleh karena itu, peneliti membuat sistem yang memanfaatkan teknologi kecerdasan buatan menggunakan algoritma *You Only Look Once* (YOLO) v8 dan ResNet-50 untuk mendeteksi dan mengklasifikasikan sampah pesisir di Pantai Bunga, Kabupaten Batubara ke dalam 3 kelas, yaitu organik, anorganik, dan bahan berbahaya beracun.

## 1.3 Batasan Masalah

Penelitian ini memiliki ruang lingkup yang dibatasi oleh beberapa hal sebagai berikut:

1. *Dataset* berupa gambar sampah yang berada di kawasan pesisir pantai dari hasil foto dengan *handphone* dan dengan jumlah yang terbatas, oleh karena itu akan dilakukan proses *augmentasi* untuk memperbanyak jumlah *dataset*.
2. Algoritma yang digunakan adalah algoritma YOLOv8 dan ResNet-50.

3. *Input* pada *website* berupa gambar yang terdiri dari 1 jenis sampah dengan posisi *full screen* dan jelas. *Outputnya* berupa informasi jenis sampah yang terdeteksi termasuk ke dalam jenis organik, anorganik, dan B3.
4. Sampah yang bisa dideteksi dan diklasifikasi yaitu sampah daun, kayu, sampah buah, plastik, *styrofoam*, kertas atau tisu, dan rokok.

#### **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah mengimplementasikan algoritma YOLOv8 untuk deteksi objek sampah dan ResNet-50 sebagai metode klasifikasi sampah pesisir Pantai Bunga di daerah Kabupaten Batubara. Peneliti juga akan mengukur performa dari implementasi algoritmanya dan melakukan evaluasi sehingga dapat menghasilkan informasi dalam menentukan jenis sampah organik, anorganik, dan sampah B3.

#### **1.5 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini yaitu :

1. Memberikan informasi mengenai penerapan algoritma *You Only Look Once* (YOLO) dan metode *Residual Network* (ResNet-50).
2. Memberikan informasi dan pengetahuan seputar jenis sampah, yaitu sampah organik, sampah anorganik, dan bahan berbahaya beracun (B3).
3. Dapat dijadikan sebagai referensi dalam menggunakan algoritma YOLO dan metode ResNet-50.
4. Dapat membantu dalam menjaga kebersihan lingkungan khususnya daerah pesisir Pantai Bunga Kabupaten Batubara.

#### **1.6 Metodologi Penelitian**

Beberapa metode yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Studi Pustaka

Tahap awal penelitian ini yaitu studi pustaka untuk melakukan pencarian referensi dari berbagai sumber terpercaya. Peninjauan pustaka dilakukan melalui buku-buku, *e-book*, artikel ilmiah, makalah ataupun situs internet yang berhubungan dengan algoritma *You Only Look Once* (YOLO) dan ResNet-50.

Hal ini bertujuan untuk memahami landasan teoritis dan kerangka konseptual yang mendukung pengembangan model YOLOv8 dan ResNet-50.

## 2. Pengumpulan *Dataset*

Langkah berikutnya adalah mengumpulkan *dataset* citra digital yang mencakup berbagai jenis-jenis gambar sampah yang tergolong ke dalam sampah organik, anorganik, dan B3. *Dataset* ini akan menjadi dasar untuk melatih dan menguji model YOLOv8 dengan ResNet-50 agar dapat mendeteksi dan mengklasifikasi jenis-jenis sampah.

## 3. Preprocessing Data

Pada tahap *pre-processing* data, *dataset* yang telah dikumpulkan akan dinormalisasi dan diubah ukurannya untuk mempersiapkan data pelatihan. Dalam *pre-processing* juga dilakukan proses anotasi data untuk menandai gambar dengan memberikan kotak pembatas dan nama kelas menggunakan *website roboflow*. Proses *augmentasi* data juga akan dilakukan untuk meningkatkan keberagaman dan generalisasi model.

## 4. *You Only Look Once v8*

Pada tahap ini, model YOLOv8 akan diimplementasikan untuk mendeteksi objek sampah. Proses pelatihan model YOLOv8 akan dilakukan menggunakan *dataset* sampah yang telah diproses sebelumnya, sehingga model dapat belajar mengenali berbagai jenis sampah dengan akurat.

## 5. *Residual Network-50*

Pada tahap ini, model ResNet-50 akan diimplementasikan untuk proses klasifikasi objek yang terdeteksi dari proses sebelumnya. Proses pelatihan model ResNet-50 juga menggunakan *dataset* yang telah diproses sebelumnya, sehingga model ResNet-50 dapat belajar dan mengklasifikasikan berbagai jenis sampah ke dalam golongan organik, anorganik dan sampah B3 dengan akurat.

## 6. Evaluasi Model

Pada tahap ini dilakukan evaluasi model dengan menggunakan *evaluation matrix* seperti: *average precision*, *precision*, *recall* dan *F1-Score* pada *training* model dan *testing* model. Evaluasi ini bertujuan untuk memastikan bahwa model mampu memberikan hasil yang akurat dan efisien.

## 7. Implementasi Sistem

Implementasi sistem dibangun sesuai dengan perancangan yang telah dibuat. Hasil akhirnya adalah dalam bentuk *website*.

## 8. Pengujian Sistem

Sistem yang telah dibuat akan di uji coba untuk melihat dan memastikan bahwa sistem tersebut berjalan sesuai dengan rancangan awal. Serta melakukan analisis untuk menilai akurasi model dalam proses deteksi dan klasifikasi sampah pesisir pantai.

## 9. Dokumentasi dan Penyusunan Laporan

Pada tahap ini, peneliti melakukan dokumentasi pada setiap tahap yang terjadi dalam penelitian dan membuat kesimpulan akhir dalam bentuk laporan penelitian atau skripsi.

### 1.7 Penelitian Relevan

Beberapa penelitian terdahulu yang relevan dengan penelitian yang dilakukan dalam penelitian ini, antara lain:

1. Berdasarkan penelitian yang dilakukan oleh Muhammad Rizki Atria Salim dkk yang berjudul “Pendeteksi Masker pada Gambar Menggunakan Model *Deep Learning* Yolo-v2 dengan ResNet-50”. Pada penelitian ini sistem deteksi masker dikembangkan menggunakan model *deep learning* YOLOv2 dengan bantuan ResNet-50. ResNet-50 digunakan sebagai *backbone layer* pengganti YOLOv2, sedangkan YOLOv2 menjadi komponen utama pendekripsi *face mask*. Berdasarkan hasil pengujian pada penelitian ini, model YOLOv2 dengan *backbone* ResNet-50 dapat digunakan sebagai alat pendekripsi masker pada citra gambar dengan memanfaatkan *face mask dataset* dan *medical mask dataset* sebagai data *training*, *validation* dan *testing*. Selain itu, pengubahan *learning rate* menjadi 0,003, *batch* menjadi 256 dan *subdivision* menjadi 32 saat melakukan *training* memiliki dampak yang cukup signifikan dalam peningkatan performansi model. Peningkatan performansi model tersebut dapat melampaui nilai pada model Yolo-v2 Original pada *testing performance*. Penelitian ini dapat memberikan skenario penggantian arsitektur Yolo-v2 dengan *backbone* ResNet-50. Penelitian ini menghasilkan nilai *F1-Score* tertinggi sebesar 84%.

2. Berdasarkan penelitian yang dilakukan oleh Kevin Reynaldi Tanjung dkk yang berjudul “Klasifikasi Benda Organik dan Anorganik Dengan Metode YOLOv3 dan ResNet50”. Metode yang digunakan yaitu YOLOv3 pada proses objek deteksi untuk mendapatkan objek sampah dari sebuah gambar atau video. Objek yang terdeteksi akan di potong dan hasilnya akan diolah oleh *Convolutional Neural Network* dengan model arsitektur ResNet50 untuk proses klasifikasi. Dalam tahap pelatihan YOLOv3 dan ResNet-50, penelitian dilakukan untuk menyesuaikan parameter agar dapat mencapai tingkat akurasi yang optimal. Fokus penelitian ini adalah pada klasifikasi objek, khususnya objek sampah, dalam gambar. Hasil penelitian menunjukkan bahwa YOLOv3 mencapai *mean average precision* sebesar 45% dengan rata-rata kehilangan (*loss*) sebesar 91%. Di sisi lain, dalam penggunaan ResNet-50, terdapat aturan praktis yang menyarankan penggunaan ukuran *input* 416 x 416 dan menurunkan *learning rate* untuk meningkatkan akurasi. Ketika kedua model ini digabungkan, ResNet-50 memiliki kemampuan untuk meningkatkan akurasi dalam mendeteksi jenis objek yang teridentifikasi oleh YOLOv3.
3. Berdasarkan penelitian Zian Md Afique Amin dkk yang berjudul “An Approach of Classifying Waste Using Transfer Learning Method”. Menggunakan beberapa metode *transfer learning* untuk melakukan proses klasifikasi. Metode yang digunakan yaitu *Xception*, *DenseNet121*, *Resnet-50*, *MobilenetV2*, dan *EffiecientNetB7*. *DenseNet121* mencapai akurasi rata-rata sekitar 93,3%. Peneliti juga menerapkan metode deteksi objek real-time seperti menggunakan YOLO atau menggunakan pendekatan serupa perangkat mikroprosesor atau kamera. *Dataset* yang digunakan terdapat tujuh kategori yang terdiri dari logam, plastik, kaca, kertas, karton, sampah, dan limbah elektronik.

## 1.8 Sistematika Penulisan

Sistematika penulisan skripsi yang digunakan dalam penelitian ini adalah sebagai berikut:

### BAB 1 PENDAHULUAN

Bab ini mencakup penjelasan mengenai latar belakang pemilihan judul, rumusan dan batasan masalah, tujuan, manfaat, dan metodologi penelitian, penelitian relevan, dan sistematika penulisan skripsi.

### BAB 2 LANDASAN TEORI

Bab ini menjelaskan beberapa teori yang berkaitan dengan penelitian, seperti analisis teoritis mengenai hal *You Only Look Once* dan *Residual Network 50*.

### BAB 3 ANALISIS DAN PERANCANGAN

Bab ini menjelaskan mengenai analisis pada algoritma dan dilakukan perancangan sistem dengan menggunakan algoritma *You Only Look Once* dan *Residual Network 50*. Serta diimplementasikan pada website berbasis *HTML*, *CSS*, dan *JavaScript*.

### BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi penjelasan mengenai implementasi algoritma mencakup pelaksanaan dan testing sistem dimana didasarkan pada langkah-langkah analisis perancangan dan sistem yang telah disebutkan diatas.

### BAB 5 KESIMPULAN DAN SARAN

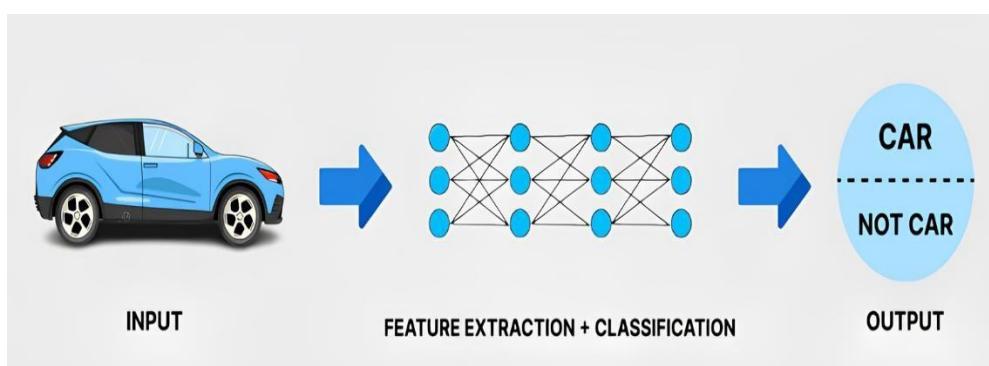
Bab ini berisi kesimpulan yang dapat diperoleh berdasarkan pemaparan pada setiap bab serta saran yang diberikan peneliti sebagai masukan untuk penelitian selanjutnya.

## BAB 2

### LANDASAN TEORI

#### 2.1 Deep Learning

*Deep Learning* adalah salah satu metode yang termasuk dalam ranah kecerdasan buatan (AI) yang bertujuan untuk mengajarkan komputer cara memproses data dengan cara yang terinspirasi oleh cara kerja otak manusia. Dengan menggunakan model-model pembelajaran yang dalam (*deep learning*), komputer mampu mengidentifikasi dan menganalisis pola-pola yang kompleks dalam berbagai jenis data seperti gambar, teks, audio, dan data lainnya. Melalui proses ini, *deep learning* mampu menghasilkan wawasan yang mendalam serta prediksi yang akurat terhadap berbagai masalah dan situasi (Fachrizal, 2021).



*Gambar 2.1 Ilustrasi Deep Learning*

Pada Gambar 2.1 dapat dilihat ilustrasi *Deep Learning*, gambar mobil sebagai data *inputan* yang ingin diklasifikasikan sebagai mobil atau bukan mobil, lalu proses ekstraksi fitur, jaringan saraf *deep learning* mengekstrak fitur penting dari gambar mobil, seperti bentuk, warna, tekstur, dan lainnya. Lalu jaringan saraf menggunakan fitur-fitur tersebut untuk mengklasifikasikan gambar sebagai mobil atau bukan mobil. Pada bagian *output*, jika jaringan saraf yakin gambar tersebut adalah mobil, maka *outputnya* akan menunjukkan "Mobil". Tetapi jika jaringan saraf yakin gambar tersebut bukan mobil, maka *outputnya* akan menunjukkan "Bukan Mobil".

## 2.2 Object Detection

*Object Detection* merupakan salah satu teknik dalam bidang penglihatan komputer yang memungkinkan kita untuk mengidentifikasi dan menemukan objek yang ada dalam gambar atau video. Dengan menggunakan teknik deteksi, kita dapat melakukan perhitungan mengenai jumlah objek yang terdapat dalam suatu pemandangan, serta menentukan dan melacak posisi masing-masing objek dengan presisi yang tinggi. Selain itu, proses ini juga memungkinkan kita untuk memberikan label secara akurat terhadap objek yang terdeteksi, membuka berbagai kemungkinan penerapan yang luas dalam berbagai bidang (Reynaldi Tanjung & Juwiantho, 2022).

Proses kerja dalam *object detection* dimulai dengan *encoder* yang menerima gambar sebagai *input* dan menjalankannya melalui serangkaian blok dan lapisan untuk mempelajari ekstraksi fitur statistik. Fitur-fitur ini nantinya digunakan untuk melakukan pencarian dan pelabelan terhadap objek dalam gambar. Hasil keluaran dari *encoder* kemudian disalurkan ke *decoder*, yang bertugas untuk memprediksi kotak pembatas dan label untuk setiap objek yang terdeteksi. Penting untuk dicatat bahwa dalam proses ini, *Intersection over Union* (IoU) digunakan sebagai kriteria untuk menentukan apakah hasil prediksi yang diperoleh adalah *true positive* atau *false positive*. *IoU* mengukur jumlah tumpang tindih antara bounding box yang mengelilingi objek yang diprediksi dan *bounding box* yang mengelilingi objek sebenarnya dalam data *ground truth*.

## 2.3 You Only Look Once (YOLOv8)

Awal kisah YOLO pada bulan Juni 2016 di Konferensi CVPR yang diadakan di Caesar's Palace Conference Center di Las Vegas, Nevada. Joseph Redmon naik panggung dan mempresentasikan makalahnya “*You Only Look Once: Unified, Real-Time Object Detection*”. Pada bulan Juli 2017 di CVPR, Joseph Redmon dan Ali Farhadi mempresentasikan makalah berjudul “YOLO9000: Lebih Baik, Lebih Cepat, Lebih Kuat” dan memperkenalkan sistem deteksi objek yang mampu mengidentifikasi lebih dari 9000 kategori. Pada tanggal 8 April 2018, Joseph Redmon dan Ali Farhadi menerbitkan

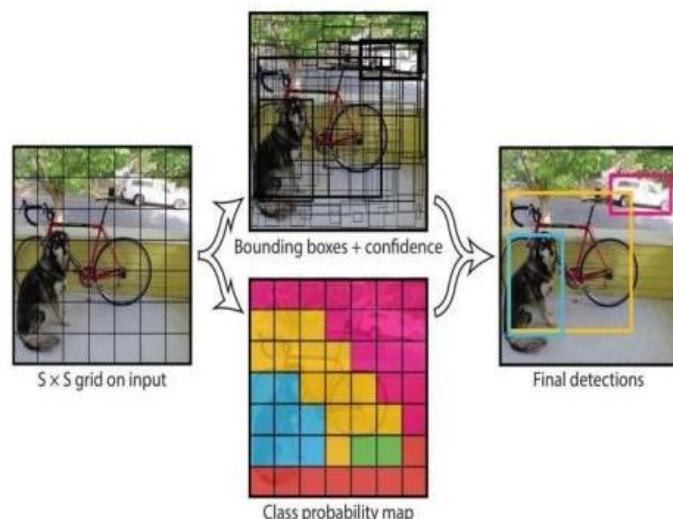
makalah tentang arXiv berjudul "YOLOv3: An Inkremental Improvement". Joseph Redmon meninggalkan penelitian visi komputer karena dia prihatin dengan aplikasi militer dan pelanggaran privasi. Pada bulan April 2020, Alexei Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao melanjutkan apa yang ditinggalkan Redmon dan merilis makalah tentang arXiv berjudul "YOLOv4: Kecepatan Optimal dan Akurasi Deteksi Objek". Pada bulan Juni 2020, hanya dua bulan setelah peluncuran YOLOv4, Glenn Jocher dari *Ultralytics* merilis YOLOv5. YOLOv5 dibuat berdasarkan banyak perbaikan yang dilakukan di YOLOv4, namun perbedaan utamanya adalah YOLOv5 dikembangkan menggunakan *PyTorch*, bukan *DarkNet*. YOLOv6 diterbitkan di ArXiv pada September 2022 oleh Departemen Meituan Vision AI. Pada bulan Juli 2022, penulis YOLOv4 dan YOLOR juga menerbitkan YOLOv7 di ArXiv. *Ultralytics*, perusahaan di balik YOLOv5, merilis YOLOv8 pada Januari 2023.

*You Only Look Once* adalah *detector* yang dapat belajar untuk melakukan deteksi pada banyak objek secara bersamaan. Algoritma ini dapat mendeteksi, mengenali dan mengklasifikasi banyak objek pada gambar dalam waktu *real-time*. YOLOv8 memungkinkan prediksi menggunakan *bounding box*. Penelitian lain membuktikan bahwa algoritma seri YOLO dapat digunakan atau diimplementasikan untuk menyelesaikan tugas pada berbagai hal skenario deteksi (Lin et al., 2021). YOLO melakukan deteksi objek dengan menggunakan *unified* model dimana sebuah *single convolutional network* memprediksi beberapa *bounding boxes* (kotak pembatas) serta probabilitas kelas di dalam kotak-kotak tersebut secara bersamaan.

Langkah awal dalam penggunaan sistem YOLO adalah membagi citra *input* ke dalam *grid* dengan ukuran  $S \times S$ . Apabila pusat dari suatu objek terletak di dalam salah satu sel *grid*, maka sel tersebut bertanggung jawab untuk mendeteksi objek tersebut. Di setiap sel *grid*, sistem memprediksi *bounding boxes* dan *confidence score* dari masing-masing *bounding box* yang dihasilkan. *Confidence score* mengindikasikan seberapa yakin dan akuratnya model dalam mendeteksi keberadaan objek di dalam kotak tersebut. Tiap *bounding box* terdiri dari lima prediksi: *x*, *y*, *w*, *h*, dan *confidence*. Koordinat

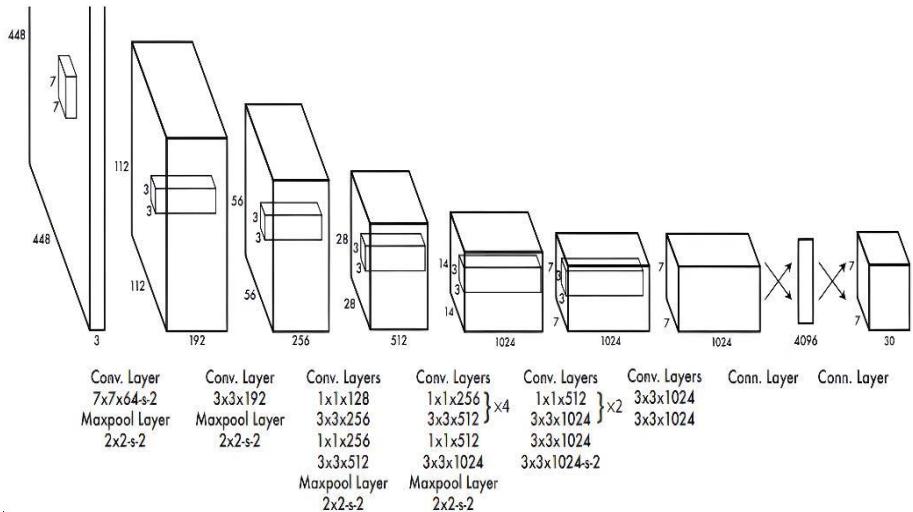
( $x, y$ ) menunjukkan posisi pusat kotak relatif terhadap batas sel *grid*, sedangkan ( $w, h$ ) menunjukkan lebar dan tinggi kotak relatif terhadap gambar secara keseluruhan.

Pada tahap akhir, *confidence* yang dihasilkan mencerminkan nilai *Intersection over Union* (IoU) antara kotak prediksi dan kotak *ground-truth*. Di samping itu, di setiap sel *grid*, sistem juga memprediksi probabilitas kelas. Probabilitas ini dikondisikan terhadap keberadaan objek di dalam sel *grid* tersebut, dan penting untuk dicatat bahwa hanya satu kelas probabilitas yang dideteksi per sel *grid*, tanpa memperhitungkan jumlah *bounding boxes*. Selain itu YOLOv8 meningkatkan *average precision* dengan *batch normalization*. Hasil prediksi pada gambar dengan model YOLO-v8 adalah *bounding box*, *confident score* dan label *class*.



**Gambar 2.2 YOLO Real Time Object Detection**

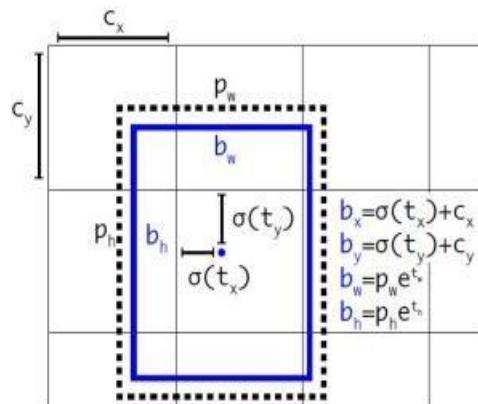
Sistem YOLO dalam pendeksiannya digunakan sebagai *classifier* dan *localizer*. YOLO menetapkan sebuah citra gambar sebagai sebuah pendeksiian ketika terdapat daerah pada citra dengan nilai *IoU score* tertinggi. *IoU* pada deteksi objek adalah mengevaluasi tingkat jarak ataupun *overlap* antara *ground truth* dan *prediction bounding box*.



**Gambar 2.3** Arsitektur You Only Look Once

### 2.3.1 Bounding box

*Bounding box* atau kotak pembatas pada sistem deteksi objek bertujuan untuk mendeteksi posisi dan jenis beberapa objek yang ada di dalam citra gambar. Pada implementasinya, *bounding box* berupa kotak-kotak pembatas yang ada pada sekitaran objek dan berbentuk persegi. Fungsinya adalah menentukan posisi, kelas, dan tingkat keyakinan (besar kemungkinan objek berada di lokasi).

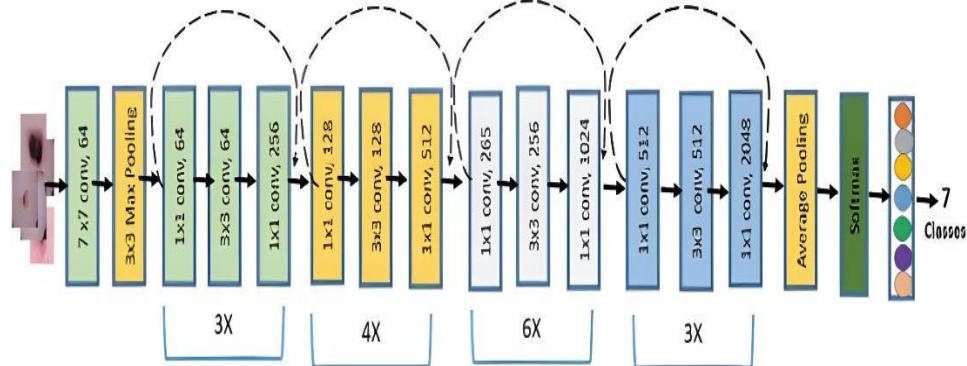


**Gambar 2.4** Prediksi Lokasi Bounding Box

### 2.4 Residual Network 50

*Residual Network* atau ResNet-50 adalah salah satu varian ResNet-50 yang dapat bekerja pada 50 lapisan *neural network*. Pada visi komputer, arsitektur ini dapat bekerja sebagai *image classification*, *object localization*, *object detection*.

*detection. Residual block* terdiri dari dua atau tiga *sequential convolutional layers* yang dalam satu *residual block* memiliki *size* 1x1, 3x3 dan 1x1 dan *shortcut layer* atau *skip connection* yang menghubungkan *input* dari lapisan pertama dan *outputnya* yang melewati beberapa lapisan model (Salim et al., 2023).



Gambar 2.5 Arsitektur ResNet-50

*Residual block* pertama diulang hingga tiga kali dan secara berurutan *filters* adalah 64, 64 dan 256. *Residual block* kedua diulang hingga empat kali dengan *filters* 128, 128 dan 512. *Residual block* ketiga diulang hingga enam kali dengan 256, 256 dan 1024. *Residual block* kedua diulang hingga empat kali dengan *filters* 128, 128 dan 512. *Residual block* keempat diulang hingga tiga kali dengan *filters* 512, 512 dan 2048.

## 2.5 Sampah

Menurut *World Health Organization* (WHO) sampah adalah suatu yang tidak digunakan, tidak dipakai, dan tidak disenangi atau sesuatu yang dibuang yang berasal dari kegiatan manusia dan tidak akan terjadi dengan sendirinya. Sampah merupakan sesuatu yang dihasilkan dari kegiatan manusia lalu sudah tidak digunakan dan dibuang (Djongghi et al., n.d.).

Dampak sampah di pesisir tidak hanya terbatas pada sektor ekonomi dan pariwisata, tetapi juga merambah ke gangguan pada ekosistem laut dan lingkungan hidup secara keseluruhan, termasuk ekosistem perairan. Keberadaan sampah juga berpotensi membahayakan kesehatan manusia, menciptakan masalah serius yang membutuhkan perhatian dan tindakan preventif serta penanganan yang komprehensif (Ayunda et al., 2023).

### 2.5.1 Organik

Sampah organik merupakan limbah atau sampah dari sisa makhluk hidup yang mudah mengalami pembusukan. Contohnya yaitu sampah buah jeruk, apel, kulit pisang, kayu, daun, bunga, dan lain-lain.



**Gambar 2.6** Sampah Organik

### 2.5.2 Anorganik

Sampah anorganik merupakan sampah yang sulit terurai atau sulit membusuk dan membutuhkan waktu yang sangat lama untuk terurai. Contohnya kertas nasi, tisu, *styrofoam*, botol minum kemasan, dan plastik.



**Gambar 2.7** Sampah Anorganik

### 2.5.3 Bahan Berbahaya dan Beracun (B3)

Sampah B3 merupakan sampah bahan berbahaya beracun (B3) yang artinya sampah yang mengandung zat kimia dan mengandung racun. Contohnya yaitu puntung rokok, baterai, kaleng obat nyamuk, obat-obatan, dan lain-lain.



*Gambar 2.8 Sampah B3*

## 2.6 Sumber Daya Kelautan Berkelanjutan

Pengelolaan yang tepat dan sesuai dengan regulasi yang berlaku di kawasan pantai memiliki potensi untuk memberikan kesuksesan, terutama dalam pengelolaan masalah sampah di area pesisir pantai tersebut. Pentingnya perhatian yang serius terhadap pengelolaan sampah tidak bisa diabaikan, mengingat risiko bencana yang dapat timbul khususnya di lingkungan pesisir. Ketika pengelolaan sampah dilakukan secara benar dan sesuai dengan ketentuan yang berlaku, dapat meningkatkan daya tarik pantai bagi pengunjung secara signifikan.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis**

Analisis adalah proses memecahkan dan menguraikan suatu informasi menjadi komponen yang lebih kecil sehingga mudah dipahami (Mulyanto, Herfandi, & Kirana, 2022). Proses analisis merupakan landasan awal sebelum melakukan perancangan dan pengembangan suatu sistem, agar sesuai dengan kebutuhan dan lebih terstruktur dalam mencapai tujuan akhir. Analisis sistem terdiri dari analisis masalah, analisis kebutuhan, dan analisis proses.

##### *3.1.1 Analisis masalah*

Analisis masalah merupakan proses dalam mendekripsi dan memahami suatu permasalahan serta mencari dan mengumpulkan data yang terkait dengan masalah tersebut agar masalah dapat diselesaikan. Penelitian ini mendekripsi permasalahan pada penggolongan kategori sampah di daerah pesisir pantai untuk membantu dalam menjaga kelestarian kawasan pantai juga membantu dalam mengoptimalkan sumber daya kelautan berkelanjutan. Harapannya masyarakat dapat mengetahui dan membedakan sampah ke dalam 3 kategori, yaitu sampah organik, anorganik, dan B3. Maka dari itu dibutuhkan sebuah sistem yang dapat mendekripsi jenis sampah dan mengklasifikasikannya ke dalam 3 kelas, yaitu sampah organik, anorganik, dan B3.

##### *3.1.2 Analisis data*

Penelitian ini menggunakan *dataset* langsung dari hasil penelitian untuk melatih model YOLO dan ResNet-50 dalam melakukan proses deteksi dan klasifikasi sampah pesisir.

Untuk *dataset* sampah yang digunakan berasal dari foto langsung menggunakan *handphone* di kawasan pesisir Pantai Bunga Kabupaten Batubara. Total *dataset* berjumlah 888 gambar.

Dari 888 gambar untuk YOLO dibagi menjadi 7 kelas, yaitu plastik sebanyak 265 gambar, *styrofoam* sebanyak 113 gambar, kertas atau tisu sebanyak 109 gambar, daun sebanyak 158 gambar, kayu sebanyak 113 gambar, sampah buah sebanyak 148 gambar, dan yang terakhir rokok sebanyak 122 gambar.

### 3.1.3 Analisis kebutuhan

Analisis kebutuhan merupakan tahapan penting yang berfokus pada pengenalan dan pemahaman kebutuhan yang diperlukan untuk merancang sistem dalam memenuhi tujuan. Analisis kebutuhan dibagi dalam dua bagianutama, yaitu fungsional dan non-fungsional.

#### 1. Kebutuhan fungsional

Kebutuhan fungsional adalah seluruh proses yang harus dilakukan dan dimiliki oleh sistem, agar dapat berfungsi dengan efektif dan akurat. Kebutuhan fungsional merupakan spesifikasi fungsionalitas yang harus ada pada suatu sistem dalam mencapai tujuan utama. Penelitian ini memiliki kebutuhan fungsional utama, yaitu:

- a. Sistem dapat membaca *input* gambar dari pengguna.
- b. Sistem mampu melakukan deteksi dengan YOLO dan klasifikasi dengan ResNet-50 terhadap *inputan* gambar sampah pesisir pantai.
- c. Sistem mampu menampilkan hasil yang baik dari deteksi gambar oleh YOLO dan klasifikasi dengan ResNet-50.

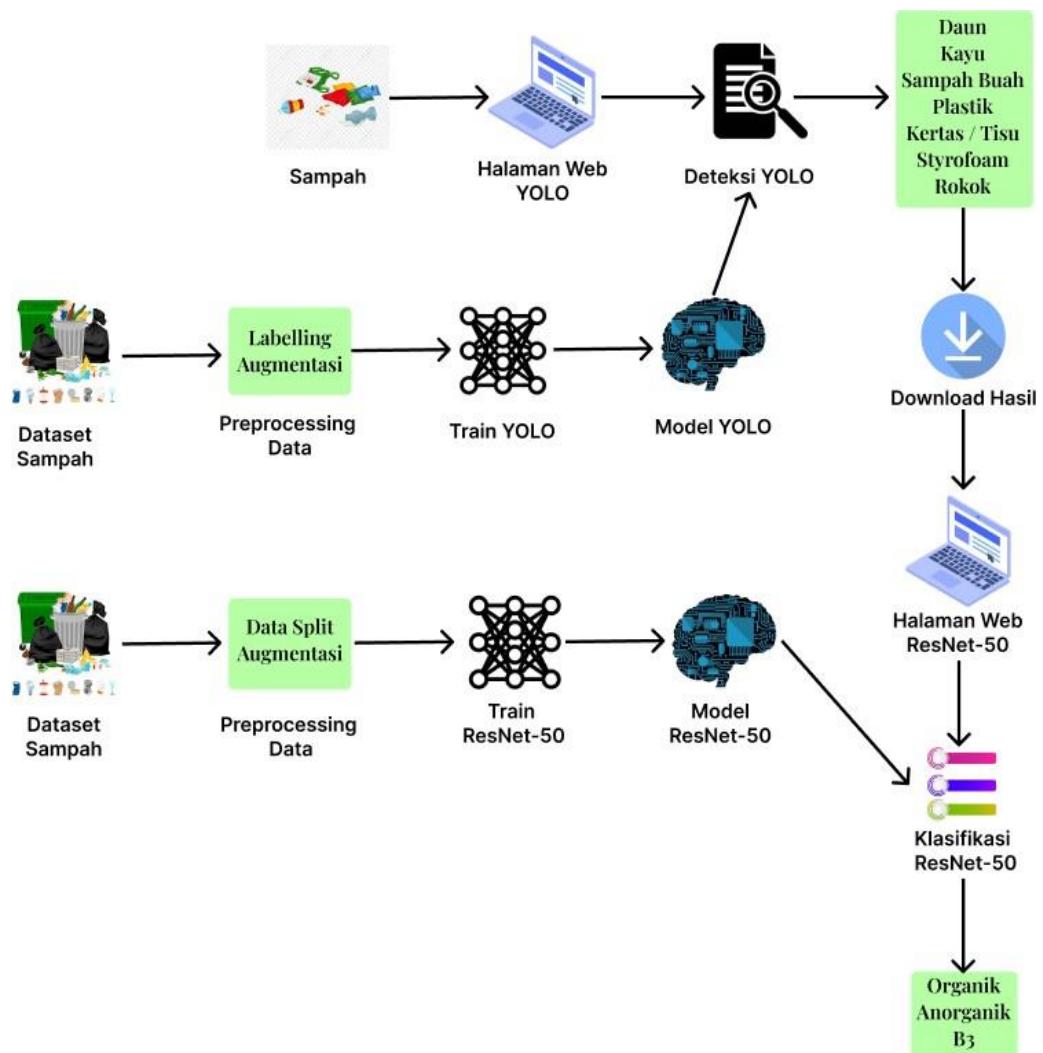
#### 2. Kebutuhan non-fungsional

Kebutuhan non-fungsional adalah spesifikasi tambahan yang mendukung sistem, dapat berupa kinerja, keamanan, serta batasan dari sistem. Penelitian ini memiliki beberapa kebutuhan non-fungsional, yaitu:

- a. Sistem memiliki tampilan yang *user friendly*.
- b. Memiliki fitur *loading* untuk mengetahui bahwa sistem sedang bekerja dalam melakukan tugas prediksi dan klasifikasi.
- c. Sistem memiliki fitur *download* untuk menyimpan gambar hasil prediksi dengan YOLO.

### 3.2 Arsitektur Umum

Perancangan sistem dibangun berdasarkan analisis yang telah dilakukan terhadap penelitian. Fokus utama dari perancangan ini adalah meningkatkan efisiensi dan efektivitas sistem yang melibatkan spesifikasi detail mengenai sistem yang dirancang dengan memanfaatkan bentuk diagram, termasuk bagaimana proses alur dari perancangan sistem program penelitian ini.



**Gambar 3.1** Arsitektur Umum

1. Pertama, siapkan foto *dataset* yang ingin dideteksi dan di klasifikasi.
2. Kedua, masuk ke halaman *website* prediksi YOLO.
3. Ketiga, *inputkan* 1 foto sampah yang akan dilakukan proses deteksi oleh YOLO.

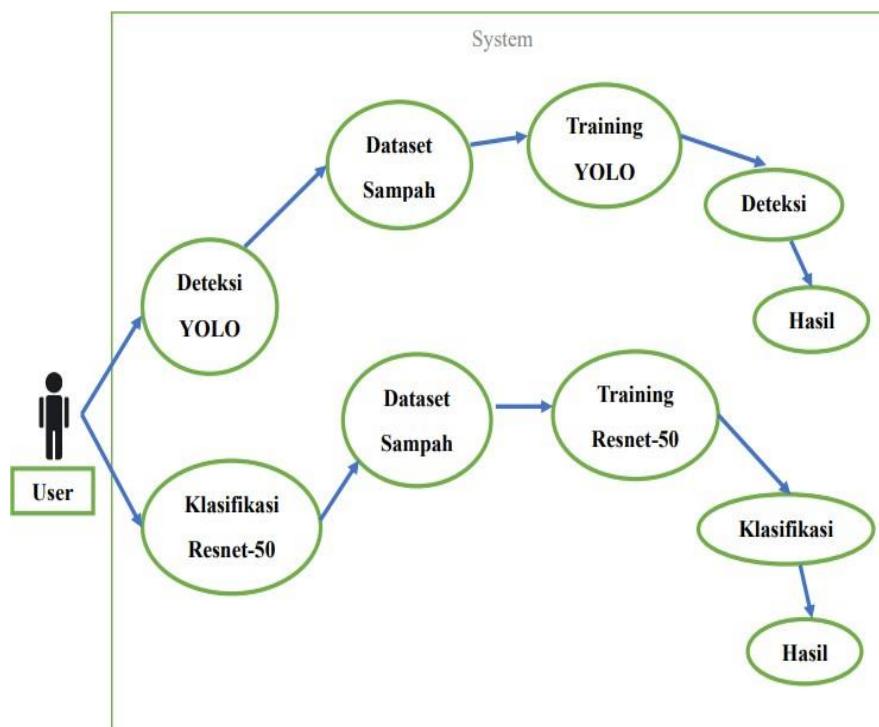
4. Hasil prediksi YOLO berdasarkan 7 kelas, yaitu sampah daun, kayu, sampah buah, plastik, *styrofoam*, kertas atau tisu, dan rokok.
5. Selanjutnya *download* hasil prediksi untuk dilakukan proses klasifikasi.
6. Masuk ke tampilan halaman *website* klasifikasi dengan ResNet-50.
7. Setelah itu, hasil prediksi oleh YOLO yang di *download* tadi, *inputkan* pada tahap ini untuk dilakukan proses klasifikasi.
8. Hasil prediksi ResNet-50 digolongkan menjadi 3 jenis sampah, yaitu Organik, Anorganik, dan B3.

### 3.3 Pemodelan Sistem

Pemodelan sistem adalah bentuk representasi dari sistem yang akan dibuat. Dalam penelitian ini, bentuk pemodelan sistem yang ditampilkan dan dijelaskan dalam bentuk diagram, yaitu *use case diagram*, *activity diagram*, *sequence diagram* beserta *flowchart* sistem.

#### 3.3.1 Use case diagram

*Use case diagram* berfungsi untuk menampilkan gambaran konsep rancangan sistem antara interaksi pengguna atau *user* dan sistem yang dirancang.

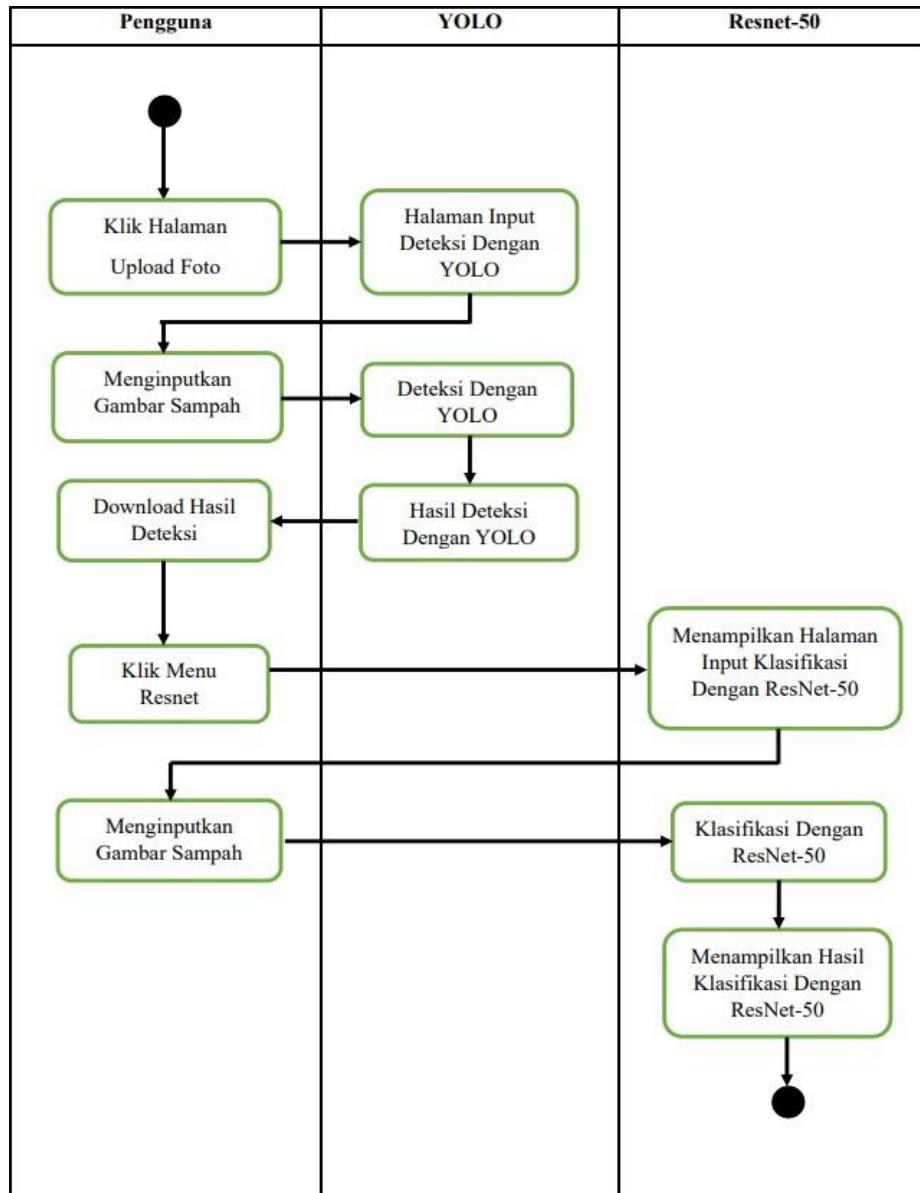


**Gambar 3.2 Use Case Diagram**

Berdasarkan Gambar 3.2, alur diagram dimulai dengan *user* membuka *website* terlebih dahulu. Lalu *user* dihadapkan pada tampilan utama *website*. Setelah itu pilih fitur “Upload Foto” disebelah kiri atas. Lalu *user* diarahkan ke halaman “Deteksi Gambar”, pada halaman deteksi gambar *user* diminta untuk memasukkan 1 foto sampah yang akan dideteksi. Setelah itu sistem akan *loading* melakukan proses deteksi dengan model YOLO. Setelah hasilnya keluar, pilih fitur *download* untuk menyimpan gambar hasil deteksi yang akan digunakan sebagai *inputan* ResNet-50. Masuk pada fitur ResNet-50 untuk proses klasifikasi, *user* kembali diminta untuk memasukkan foto hasil deteksi yang telah di *download*. Kemudian sistem akan melakukan proses klasifikasi dengan model ResNet-50. Terakhir, sistem akan memperlihatkan hasil dari klasifikasi berupa golongan jenis sampah yang terdeteksi.

### 3.3.2 *Activity diagram*

*Activity diagram* berfungsi untuk menampilkan alur kegiatan rancangan sistem, dari awal mulai *website* dijalankan, lalu proses, dan selesai.

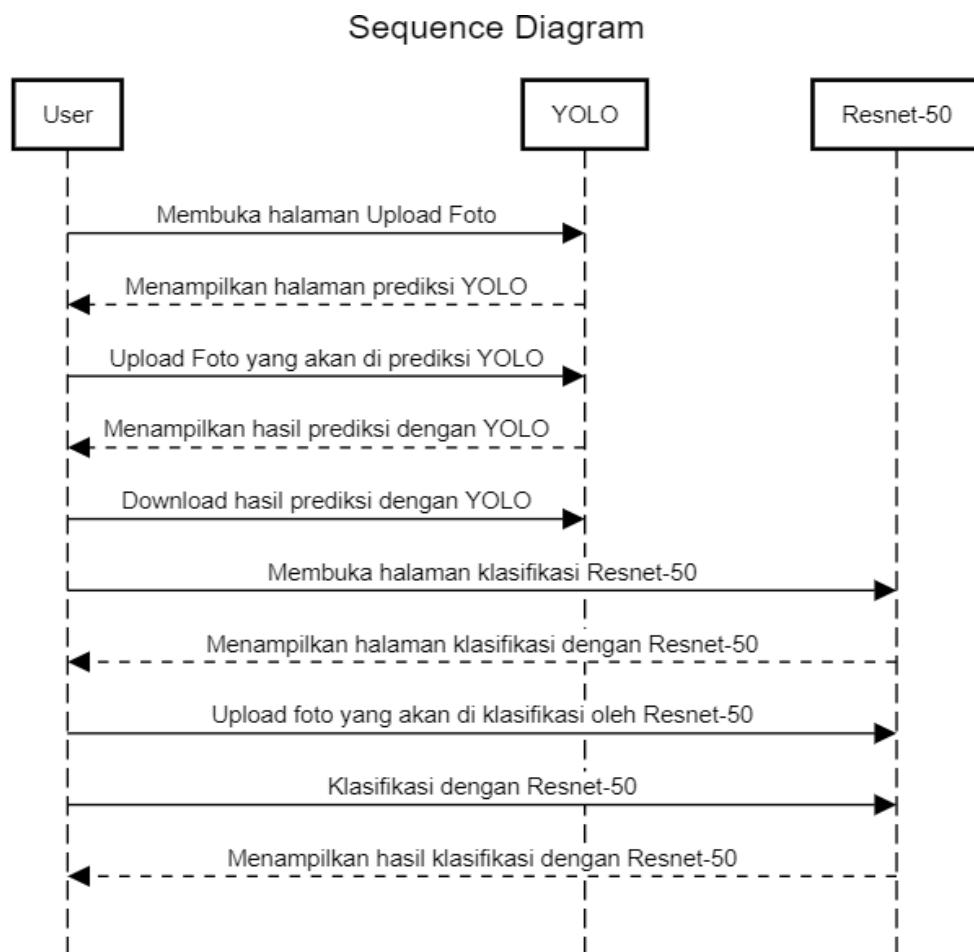


**Gambar 3.3 Activity Diagram**

Berdasarkan Gambar 3.3, alur diagram dimulai dengan *user* memilih tombol atau menu untuk *upload* foto, lalu *website* akan menampilkan halaman *input* foto untuk melakukan proses deteksi, masukkan 1 foto sampah yang akan dilakukan proses deteksi. Selanjutnya *user* menekan fitur “Deteksi” agar proses deteksi berjalan. Setelah beberapa saat hasil deteksi keluar, klik fitur *download*, lalu klik fitur ResNet-50 untuk melakukan proses klasifikasi. Setelah itu *user* menginputkan foto sampah hasil deteksi yang telah di *download*. Setelah *diinput*, klik fitur “Klasifikasi” untuk dilakukan proses klasifikasi. Terakhir sistem akan memperlihatkan hasil klasifikasi sampah berdasarkan 3 golongan, yaitu organik, anorganik, dan B3.

### 3.3.3 Sequence diagram

*Sequence diagram* bertujuan untuk memvisualisasikan urutan proses yang terjadi antara objek atau komponen dalam sistem. Diagram ini membantu kita memahami bagaimana sistem berperilaku pada saat akan digunakan.



**Gambar 3.4 Sequence Diagram**

Gambar 3.4 menjelaskan bahwa terdapat 2 komponen yaitu *user* dan *website* yang saling berinteraksi. Diagram diatas menjelaskan proses dari awal *user* membuka *website* dan dihadapkan pada tampilan utama *website* yang terdiri dari tampilan YOLO dan ResNet-50 hingga sampai proses deteksi dan klasifikasi gambar sampah selesai.

### 3.3.4 Flowchart sistem

*Flowchart* merupakan penggambaran visual dari serangkaian proses, keputusan, dan aliran logika dari suatu sistem yang direpresentasikan dalam

bentuk diagram alir dengan beberapa simbol yang ditetapkan oleh *American National Standards Institute* (ANSI), seperti panah, persegi panjang, hexagon, dan lainnya.



*Gambar 3.5 Flowchart Sistem*

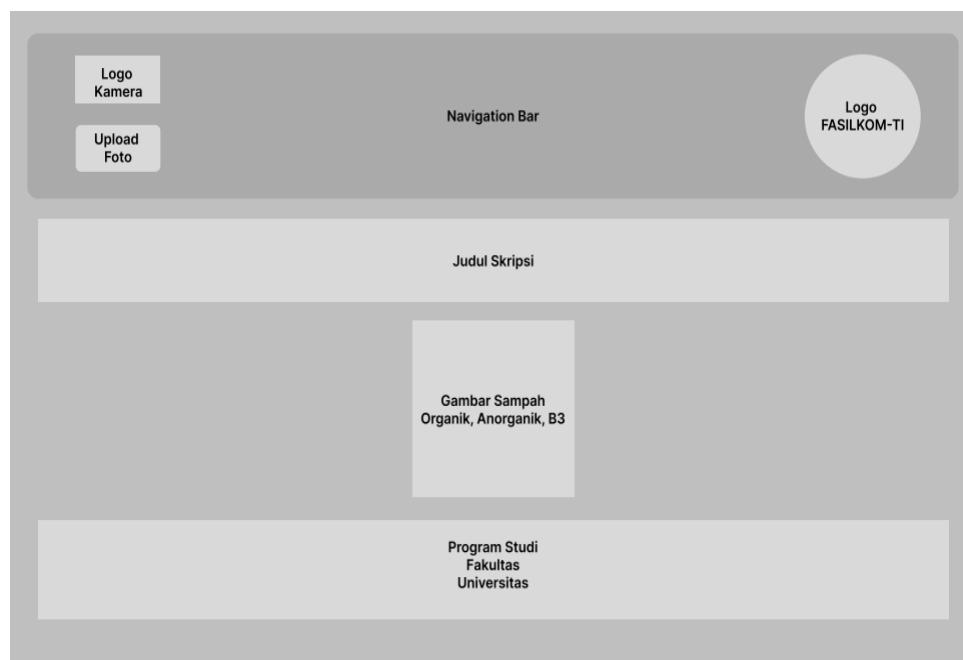
Pada Gambar 3.5 diatas menjelaskan diagram alir dari sistem, yang pertama yakni *user menginputkan* 1 gambar sampah yang akan akan didetect. Kemudian gambar tersebut dikirim ke API untuk dilakukan *preprocess* sebelum dikirim ke model. Selanjutnya model melakukan prediksi dan akan mengembalikan respons hasil prediksi kepada *user*, setelah itu *user mendownload* hasil deteksi.

Selanjutnya masuk ke halaman klasifikasi dengan ResNet-50, *inputkan* foto dari hasil deteksi yang telah di *download* tadi untuk dilakukan proses klasifikasi. Terakhir *website* menampilkan hasil *user* berupa penggolongan jenis sampah di antaranya sampah organik, anorganik, dan B3.

### 3.4 Perancangan *Interface*

Perancangan *Interface* adalah proses menciptakan kerangka atau desain sistem yang akan dibuat. Antarmuka (*Interface*) dalam sebuah sistem adalah elemen integral yang memfasilitasi interaksi antara pengguna dan sistem. Pentingnya memahami struktur sistem terkait sebelum merancang antarmuka adalah untuk memastikan implementasi yang efisien. Rancangan antarmuka harus memperhatikan kebutuhan pengguna dengan menyediakan tampilan yang sederhana dan nyaman untuk memudahkan interaksi.

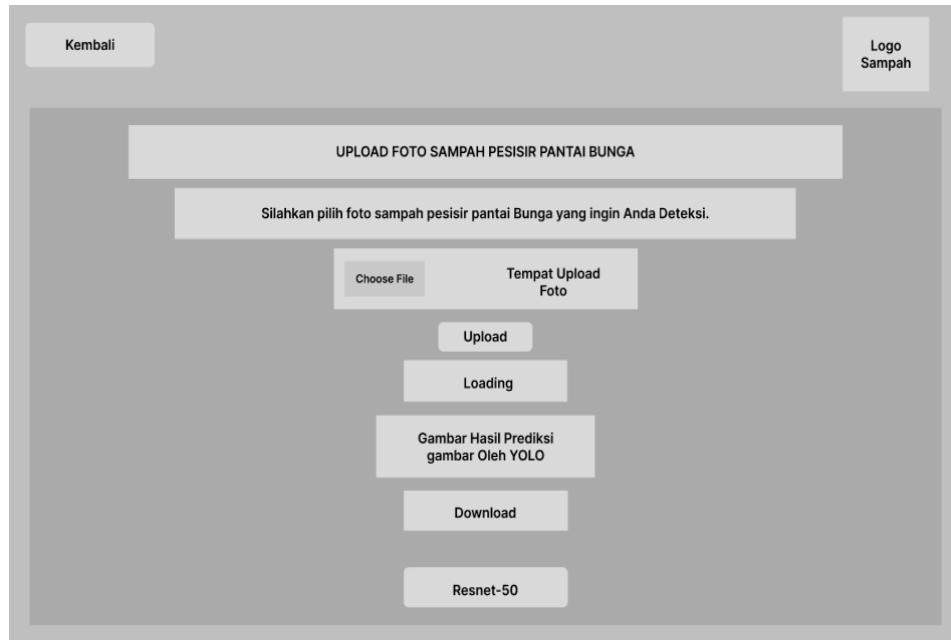
#### 3.4.1 Halaman utama website



**Gambar 3.6** Halaman Utama

Gambar 3.6 menunjukkan secara singkat bagaimana antarmuka halaman utama *website* ketika *user* pertama kali menggunakan *website*. Pengguna dapat memilih fitur “Upload Foto” untuk menggunakan *website* dalam tugas prediksi dan klasifikasi sampah.

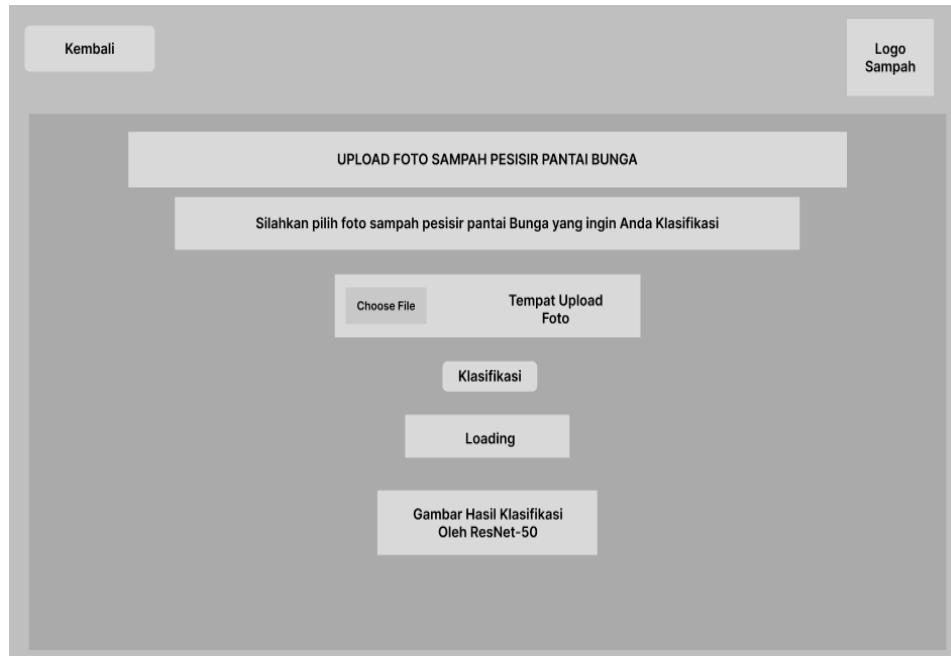
### 3.4.2 Halaman deteksi dengan YOLO



**Gambar 3.7** Halaman Deteksi YOLO

Gambar 3.7 menunjukkan secara singkat bagaimana tampilan antarmuka halaman deteksi pada *website*. Terdapat beberapa menu seperti pilih foto, *upload*, *download*, dan ResNet-50. Lalu terdapat tampilan hasil dari gambar yang telah di prediksi.

### 3.4.3 Halaman klasifikasi dengan ResNet-50



**Gambar 3.8** Halaman Klasifikasi ResNet-50

Gambar 3.8 menunjukkan secara singkat bagaimana tampilan antarmuka halaman klasifikasi pada *website*. Terdapat beberapa menu seperti pilih foto, klasifikasi dan terakhir tampilan hasil dari gambar yang telah di prediksi.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN SISTEM

#### 4.1 Implementasi Sistem

Setelah tahap analisis dan perancangan, tahap selanjutnya adalah implementasi dan pengujian, di mana dalam tahap ini penelitian melibatkan sebuah *software*, yaitu *Visual Studio Code*, *Google Colaboratory*, dan *Postman*. Bahasa pemrograman yang digunakan penelitian ini adalah Python sebagai *back-end* dan HTML, CSS, JavaScript sebagai *front-end*.

##### 4.1.1 Landing page

Gambar di bawah adalah tampilan halaman utama yang digunakan dalam sistem.



**Gambar 4.1** Halaman Utama

##### 4.1.2 Halaman upload foto YOLO

Gambar di bawah yaitu proses pendekripsi objek dengan algoritma YOLO. Dengan menginputkan gambar sampah yang ingin dideteksi. Sampah yang bisa dideteksi hanya sampah daun, kayu, sampah buah, plastik, *styrofoam*, kertas atau tisu, dan rokok.

## UPLOAD FOTO SAMPAH PESISIR PANTAI BUNGA

Silahkan pilih foto sampah pesisir pantai Bunga yang ingin Anda Deteksi.

No file chosen  
  

  
RESNET-50

**Gambar 4.2 Halaman Deteksi YOLO**

### 4.1.3 Hasil Deteksi You Only Look Once

Setelah *diinputkan*, lalu di *upload*, gambar di bawah yaitu hasil dari prediksi objek sampah oleh YOLO. Prediksi gambar berhasil dilakukan oleh YOLO dengan *output* sampah buah sebanyak 1. Setelah itu gambar di *download* untuk melakukan proses klasifikasi oleh ResNet-50.

## UPLOAD FOTO SAMPAH PESISIR PANTAI BUNGA

Silahkan pilih foto sampah pesisir pantai Bunga yang ingin Anda Deteksi.



**Gambar 4.3 Halaman Hasil Deteksi**

### 4.1.4 Proses Klasifikasi ResNet-50

Setelah gambar di *download*, lalu klik pada bagian ResNet-50, lalu masukkan gambar sampah yang di *download* tadi untuk dilakukan proses

klasifikasi. Gambar di bawah berhasil untuk mengklasifikasikan sampah yang terdeteksi oleh YOLO dengan *inputan* ‘sampah buah’ menghasilkan klasifikasi ‘Organik’.



**Gambar 4.4** Halaman Klasifikasi ResNet-50

## 4.2 Pengujian Sistem

Pada bagian ini yaitu tahapan uji coba untuk menguji sistem bahwa implementasi telah berhasil dilakukan. Pengujian sistem memiliki fungsi untuk menampilkan bahwa sistem yang diciptakan mampu bekerja dengan benar dalam mendeteksi objek sampah dan mengklasifikasikan sampah ke dalam 3 kelas, yaitu sampah organik, anorganik, dan B3. Pada tahap ini terdapat 7 jenis sampah yang dapat dideteksi, yaitu daun, kayu, sampah buah, plastik, *styrofoam*, kertas atau tisu, dan rokok. Berikut adalah tabel yang akan menampilkan kelas dan jenis sampah yang digunakan.

**Tabel 4.1** Daftar Kelas dan Jenis Sampah

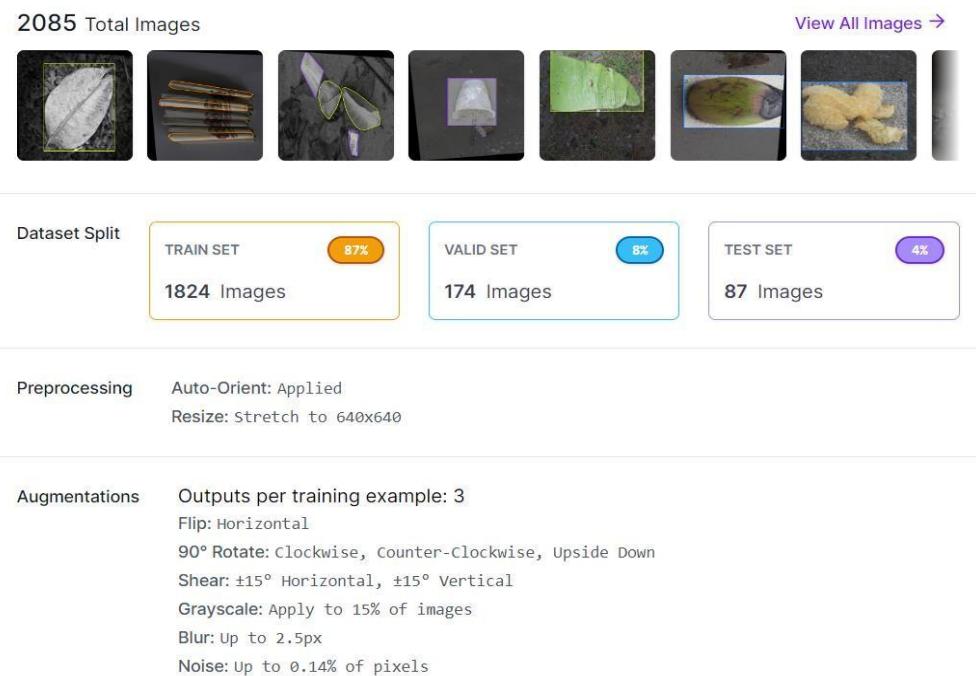
No	Kelas Sampah	Jenis Sampah
1	Organik	Daun
		Kayu
		Sampah buah
2	Anorganik	Plastik
		Kertas atau tisu

		Styrofoam
3	B3	Rokok

#### 4.2.1 Program algoritma You Only Look Once

Berikut ini adalah tahapan dalam proses pembangunan sistem, yang dimulai dengan model pelatihan model.

1. Dimulai dengan program YOLO. Setalah pengumpulan *dataset* dilakukan, *dataset* diolah di *roboflow* untuk dilakukan proses pelabelan *dataset* serta *augmentasi* dengan jumlah total *dataset* 2085.



**Gambar 4.5** Labeling dan Augmentasi di Roboflow

2. Selanjutnya *menginstall library Roboflow*. Lalu impor kelas *roboflow* dengan mengakses kunci *API* yang diberikan untuk mengambil dan mengelola proyek *dataset* untuk di proses di Google Colab.

```
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="AUYopPBMEndHGz5DmKCQ")
project = rf.workspace("universitas-sumatera-utara-dvjul").project("trash-oakg2")
dataset = project.version(1).download("yolov8")
```

**Gambar 4.6** Library

3. *Install library ultralytics* untuk membantu dalam proses YOLO bekerja.

```
!pip install ultralytics
```

*Gambar 4.7 Install Ultralytics*

4. Selanjutnya *import ultralytics* dan cek *ultralytics* apakah sudah *terinstall* atau belum.

```
import ultralytics
ultralytics.checks()
```

*Gambar 4.8 Import Ultralytics*

5. Lanjut *import* kelas YOLO dari *library ultralytics* untuk proses deteksi objek menggunakan model YOLO.

```
from ultralytics import YOLO
```

*Gambar 4.9 Import YOLO*

6. Selanjutnya menginisialisasi model YOLO dari *library ultralytics* dengan menggunakan ‘yolov8n.pt’ yang menunjukkan bahwa model YOLO yang digunakan adalah versi 8.

```
model = YOLO('yolov8n.pt')
```

*Gambar 4.10 Model YOLOv8*

7. Lanjut ke proses *training* atau pelatihan model YOLO dengan mengakses *dataset* ‘Trash-1’ dengan ukuran *batch* (jumlah sampel yang di proses dalam satu iterasi) sebanyak 10, dengan *device* GPU, dan *epoch* sebanyak 25 kali lalu nilai 640 menandakan dimensi gambar *input* setinggi 640 piksel.

```
results = model.train(data='/content/Trash-1/data.yaml',
pretrained=True,batch=10, device=0, epochs=25, imgsz=640)
```

**Gambar 4.11** Pelatihan Model

*Epoch* ke 25 dengan nilai *mAP50* sebesar 0.808 serta model *summary* yang terlihat di gambar pada setiap kelasnya. Selama *epoch* terlihat bahwa *loss* menurun seiring berjalananya *epoch*, dan metrik evaluasi seperti *precision*, *recall*, dan *mAP* meningkat, menunjukkan bahwa model semakin baik dalam mendekripsi objek pada *dataset* yang diuji.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
25/25	1.64G	0.8266	0.6945	1.302	4	640: 100% [██████████] 183/183 [00:41<00:00, 4.41it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95: 100% [██████████] 9/9 [00:02<00:00, 3.28it/s]

```
25 epochs completed in 0.326 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 6.3MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 6.3MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.1.29 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3007013 parameters, 0 gradients, 8.1 GFLOPs
    Class      Images Instances   Box(P)      R      mAP50 mAP50-95: 100% [██████████] 9/9 [00:04<00:00, 2.08it/s]
        all       174     198      0.85      0.71      0.807      0.572
        daun      174      33      0.787      0.667      0.743      0.596
        kayu      174      21      0.978      0.762      0.832      0.513
        kertas atau tisu 174      25      0.986      0.68      0.858      0.616
        plastik    174      47      0.935      0.851      0.935      0.682
        rokok      174      21      0.716      0.714      0.815      0.497
        sampah buah 174      31      0.724      0.548      0.654      0.463
        styrofoam   174      20      0.824      0.75      0.815      0.639
Speed: 0.2ms preprocess, 6.4ms inference, 0.0ms loss, 7.2ms postprocess per image
Results saved to runs/detect/train2
```

**Gambar 4.12** Progres Pelatihan

8. Import modul ‘*locale*’ dengan tujuan menetapkan preferensi *encoding* (pengkodean) default menjadi “UTF-8” menggunakan modul *locale* di *Python*. Pengaturan ini bertujuan untuk mengubah preferensi *encoding default* yang digunakan oleh beberapa operasi yang melibatkan teks atau *string*, seperti membaca atau menulis file.

```
import locale
locale.getpreferredencoding = lambda: "UTF-8"
```

**Gambar 4.13** Import Locale

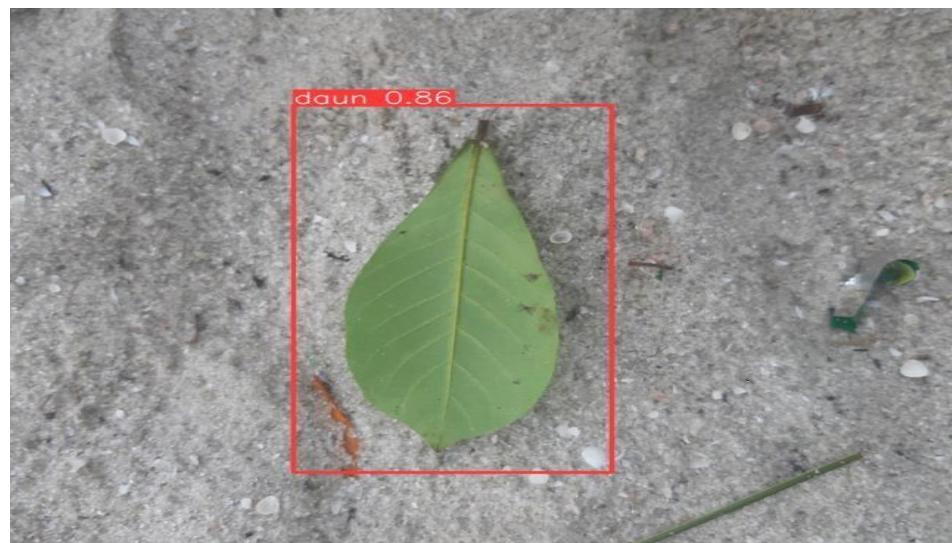
9. Selanjutnya program untuk melakukan deteksi dan prediksi objek dengan memasukkan *path* file objek dari data *valid* yang menghasilkan model ‘best.pt’ lalu nilai ‘true’ menandakan bahwa hasil akan disimpan.

```
!yolo detect predict model=/content/runs/detect/train2/weights/best.pt
source='/content/Trash-1/valid/images/20240107_091422.jpg_rf.d962b1d3e32db596d8fe48526cc8195b.jpg'
save=true
```

**Gambar 4.14** Path gambar yang dideteksi dan model ‘best.pt’

#### 10. Hasil *training*

Hasil prediksi dengan YOLO dengan nilai pada *bounding box* 0.86 yaitu sampah daun.



**Gambar 4.15** Hasil Deteksi YOLO

11. Gambar dibawah menunjukkan grafik garis dari beberapa metrik yang diukur selama proses pelatihan model YOLOv8. Grafik dibagi menjadi dua bagian: *train* (pelatihan) dan *valid* (validasi). Bagian *train* menunjukkan performa model selama proses pelatihan. Metrik yang ditampilkan adalah:

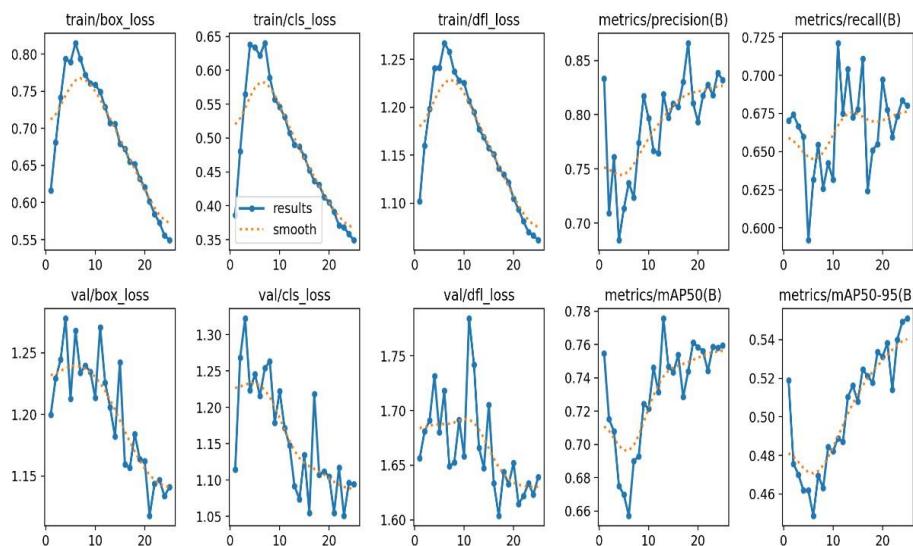
- train/box loss*: Ukuran seberapa akurat model dalam memprediksi kotak pembatas di sekitar objek.
- train/cls loss*: Ukuran seberapa akurat model dalam mendeteksi kelas objek di dalam kotak pembatas.
- train/dfl loss*: Ukuran seberapa akurat model dalam memprediksi dimensi dan lokasi kotak pembatas.

- d) *metrics/precision(B)*: Ukuran seberapa besar proporsi prediksi positif yang benar.
- e) *metrics/recall(B)*: Ukuran seberapa besar proporsi objek positif yang terdeteksi.

Bagian *val* menunjukkan performa model pada *dataset* validasi. Metrik yang ditampilkan adalah:

- a) *val/box loss*: Ukuran seberapa akurat model dalam memprediksi kotak pembatas di sekitar objek pada *dataset* validasi.
- b) *val/cls loss*: Ukuran seberapa akurat model dalam mendeteksi kelas objek di dalam kotak pembatas pada *dataset* validasi.
- c) *val/dfl loss*: Ukuran seberapa akurat model dalam memprediksi dimensi dan lokasi kotak pembatas pada *dataset* validasi.
- d) *metrics/mAP50(B)*: Ukuran rata-rata presisi rata-rata pada 50 nilai ambang batas *Intersection over Union* (IoU) yang berbeda.
- e) *metrics/mAP50-95(B)*: Ukuran rata-rata presisi rata-rata pada 50 nilai ambang batas *IoU* yang berbeda, dengan fokus pada deteksi objek yang lebih kecil.

Secara umum, grafik menunjukkan bahwa model YOLOv8 mengalami peningkatan performa selama proses pelatihan. Hal ini terlihat dari penurunan nilai *loss* dan peningkatan nilai *precision* dan *recall* pada kedua bagian *train* dan *val*.



**Gambar 4.16** Grafik Pelatihan Model

#### 4.2.2 Pengujian model YOLO

Berikut ini adalah tahapan dalam proses pengujian model dari YOLOv8. Hasil prediksi didapat dari hasil percobaan model dengan gambar sampah sebanyak 85 gambar.

**Tabel 4.2** Pengujian Model YOLO

Nama Kelas	Hasil Prediksi Sebenarnya							
	Daun	Kayu	Sampah Buah	Kertas/ Tisu	Plastik	Styrofoam	Rokok	Tidak Terdeteksi
<b>Daun</b>	14	0	0	0	0	0	0	0
<b>Kayu</b>	0	10	1	0	0	0	0	1
<b>Sampah Buah</b>	1	0	8	0	0	0	0	1
<b>Kertas/ Tisu</b>	0	0	0	5	0	0	0	1
<b>Plastik</b>	1	0	0	1	23	1	0	0
<b>Styrofoam</b>	0	0	0	1	2	6	0	0
<b>Rokok</b>	0	0	0	0	0	0	8	0

Evaluasi kinerja menggunakan *confusion matrix* memanfaatkan empat konsep untuk menggambarkan hasil dari proses klasifikasi, yaitu:

- *True Positive* (TP) = Merupakan jumlah kasus positif yang benar-benar diklasifikasikan sebagai positif oleh model.
- *True Negative* (TN) = Merupakan jumlah kasus negatif yang benar-benar diklasifikasikan sebagai negatif oleh model.
- *False Positive* (FP) = Merupakan jumlah kasus negatif yang salah diklasifikasikan sebagai positif oleh model.
- *False Negative* (FN) = Merupakan jumlah kasus positif yang salah diklasifikasikan sebagai negatif oleh model.

*Accuracy* (akurasi) adalah ukuran umum untuk mengevaluasi kinerja dari sebuah model klasifikasi. Ini mengukur seberapa sering model tersebut membuat prediksi yang benar, secara keseluruhan, dari semua kasus yang diamati. Dalam konteks *confusion matrix*, akurasi dihitung dengan rumus:

$$\text{Accuracy} = \frac{\text{Total TP}}{\text{Total Data}}$$

$$\text{Accuracy} = \frac{14+10+8+5+23+6+8}{85} = \frac{74}{85} = 0.87$$

*Precision* (presisi) adalah ukuran yang menggambarkan seberapa akurat model dalam mengklasifikasikan kasus positif. Dalam konteks *confusion matrix*, *precision* mengacu pada proporsi dari data yang secara tepat diklasifikasikan sebagai positif dari keseluruhan data yang diprediksi sebagai positif. Presisi dihitung dengan rumus:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

*Precision* untuk kelas “Daun”

$$\text{TP} = 14$$

$$\text{FP} = 0+1+0+1+0+0 = 2$$

*Precision* untuk kelas “Kayu”

$$\text{TP} = 10$$

$$\text{FP} = 0+0+0+0+0+0 = 0$$

*Precision* untuk kelas “Sampah Buah”

$$\text{TP} = 8$$

$$\text{FP} = 0+1+0+0+0+0 = 1$$

*Precision* untuk kelas “Kertas / Tisu”

$$\text{TP} = 5$$

$$\text{FP} = 0+0+0+1+1+0 = 2$$

*Precision* untuk kelas “Plastik”

$$TP = 23$$

$$FP = 0+0+0+0+2+0 = 2$$

*Precision* untuk kelas “Styrofoam”

$$TP = 6$$

$$FP = 0+0+0+0+1+0 = 1$$

*Precision* untuk kelas “Rokok”

$$TP = 8$$

$$FP = 0+0+0+0+0+0 = 0$$

$$Precision = \frac{14+10+8+5+23+5+8}{16+10+9+7+26+7+8} = \frac{74}{82} = 0.90$$

*Recall* adalah ukuran keberhasilan sistem dalam mendapatkan informasi yang relevan. Rumus untuk menghitung *recall* dapat ditemukan seperti pada rumus dibawah ini.

$$\textbf{\textit{Recall}} = \frac{\textbf{\textit{TP}}}{\textbf{\textit{TP}} + \textbf{\textit{FN}}}$$

*Recall* untuk kelas “Daun”

$$TP = 14$$

$$FN = 0+0+0+0+0+0 = 0$$

*Recall* untuk kelas “Kayu”

$$TP = 10$$

$$FN = 0+1+0+0+0+0 = 1$$

*Recall* untuk kelas “Sampah Buah”

$$TP = 8$$

$$FN = 1+0+0+0+0+0 = 1$$

*Recall* untuk kelas “Kertas / Tisu”

$$TP = 5$$

$$FN = 0+0+0+0+0+0 = 0$$

*Recall* untuk kelas “Plastik”

$$TP = 23$$

$$FN = 1+0+0+1+1+0 = 3$$

*Recall* untuk kelas “Styrofoam”

$$TP = 6$$

$$FN = 0+0+0+1+2+0 = 3$$

*Recall* untuk kelas “Rokok”

$$TP = 8$$

$$FN = 0+0+0+0+0+0 = 0$$

$$Recall = \frac{14+10+8+5+23+6+8}{14+11+10+5+26+9+8} = \frac{74}{83} = 0.89$$

*F1 Score* adalah nilai yang membandingkan antara *precision* dan *recall*. Cara menghitung *F1 Score* mengikuti rumus dibawah ini.

$$F1\ Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$F1\ Score = 2 \times \frac{0.90 \times 0.89}{0.90 + 0.89}$$

$$F1\ Score = 2 \times \frac{0.801}{1.79} = 0.89$$

#### 4.2.3 Program metode ResNet-50

Berikut ini adalah tahapan dalam proses pembangunan sistem, yang dimulai dengan pelatihan model.

1. Lanjut ke metode ResNet-50, yang pertama yaitu *import library*

```
import os
import numpy as np
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model, Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout, BatchNormalization
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input, decode_predictions
from sklearn.metrics import classification_report, confusion_matrix, f1_score, accuracy_score, roc_auc_score
```

**Gambar 4.17 Library**

Pada gambar diatas, menjelaskan hal yang pertama sekali perlu dilakukan adalah melakukan import terhadap *library* yang akan digunakan. Pada model, diperlukan beberapa *library* guna mendukung penyelesaian latihan. *Library* yang diperlukan antara lain yaitu, *os*, *numpy*, *seaborn*, *tensorflow*, *keras*, *sklearn*, dan *matplotlib*.

2. Memuat model ResNet50 yang telah dilatih sebelumnya (*pre-trained*) tanpa lapisan penuh teratas (*fully connected layers*) karena menggunakan model yang telah dilatih pada *dataset* besar (seperti *ImageNet*).

```
# Load pre-trained ResNet model without the top (fully connected) layers
resnet_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

**Gambar 4.18** Model ResNet-50

3. Proses membekukan lapisan-lapisan dari model ResNet-50 yang telah dilatih sebelumnya. Membekukan lapisan-lapisan dari model *pre-trained* umumnya dilakukan untuk memastikan bahwa representasi fitur yang telah dipelajari dari *dataset* asli (*ImageNet*) tetap utuh dan tidak terpengaruh oleh *dataset* tugas klasifikasi yang lebih kecil atau berbeda. Hal ini terutama berguna saat kita memiliki keterbatasan data pelatihan, dan kita ingin memanfaatkan pengetahuan yang luas yang telah diperoleh oleh model pada *dataset* yang lebih besar.

```
# Freeze the pre-trained layers
for layer in resnet_model.layers:
    layer.trainable = False
```

**Gambar 4.19** Membekukan Lapisan Model

4. Menghapus atau memotong lapisan terakhir (*output layer*) dari model ResNet-50. Menciptakan model yang baru dengan menggunakan semua lapisan sebelumnya dari ResNet-50, kecuali *output layer* yang dihilangkan. Setelah itu, nantinya *output layer* baru akan ditambahkan atau diganti dengan *output layer* yang sesuai dengan tugas klasifikasi sampah. Cara ini, memanfaatkan fitur tingkat tinggi yang telah dipelajari oleh model ResNet-50 tanpa perlu melatih seluruh model dari awal.

```
# Hapus lapisan terakhir (output layer) karena kita ingin menggantinya dengan output klasifikasi sampah
resnet_model = models.Model(inputs=resnet_model.input, outputs=resnet_model.layers[-2].output)
```

**Gambar 4.20** Menghapus Lapisan Terakhir

5. Selanjutnya menambahkan lapisan-lapisan klasifikasi kustom ke model yang telah diubah sebelumnya. Model telah diubah dan diperluas untuk tugas klasifikasi sampah dengan menambahkan lapisan-lapisan khusus untuk menghasilkan *output* yang sesuai dengan jumlah kelas yang diinginkan.

```
# Add custom classification layers
x = resnet_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
x = Dense(256, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.3)(x)
predictions = Dense(3, activation='softmax')(x) # Change the number of units to 3 for 3 classes

model = Model(inputs=resnet_model.input, outputs=predictions)
```

**Gambar 4.21** Menambahkan Lapisan Kustom

6. Selanjutnya Mengatur ‘*resnet\_model.trainable*’ menjadi ‘*False*’, kita memberi tahu model bahwa lapisan-lapisan tersebut tidak perlu di-*update* selama proses pelatihan.

```
# Membekukan lapisan ResNet agar tidak terlatih ulang selama pelatihan model klasifikasi sampah
resnet_model.trainable = False
```

**Gambar 4.22** Mengatur ‘*resnet\_model.trainable*’

7. Selanjutnya kompilasi model yang telah diubah dan diperluas sebelumnya untuk tugas klasifikasi sampah. Memilih algoritma optimasi “Adam” sebagai pengoptimal model. Menggunakan ‘*categorical\_crossentropy*’ karena model memiliki tiga kelas pada *output layer* dan tugas klasifikasi adalah multi-kelas. Menetapkan metrik evaluasi yang akan digunakan untuk mengukur kinerja model selama pelatihan dan evaluasi. Kemudian menggunakan *learning rate* dengan nilai 0.0001.

```
# Kompilasi model

learning_rate=0.0001

model.compile(optimizer=Adam(learning_rate=learning_rate),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

**Gambar 4.23** Kompilasi Model

8. Program di bawah bertujuan menghubungkan ke Google Drive untuk mengakses *dataset* yang tersimpan di Google Drive.

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

**Gambar 4.24** Menghubungkan ke Google Drive

9. Selanjutnya mendefinisikan sebuah *generator* data yang akan digunakan selama pelatihan model. *Generator* ini dirancang khusus untuk mengelola data gambar dari direktori tertentu dan menyediakan *batch-batch* data yang akan digunakan pada setiap iterasi pelatihan. *Outputnya* menghasilkan terdapat 693 gambar dari 3 kelas.

```
# Definisi generator data untuk pelatihan
datagen = image.ImageDataGenerator(preprocessing_function=preprocess_input)

train_generator = datagen.flow_from_directory(
    '/content/drive/MyDrive/Sampah Resnet/Train', # Ganti dengan path dataset pelatihan
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical', # Menggunakan categorical karena memiliki 3 kelas
    shuffle=True
)
```

**Gambar 4.25** Mendefenisikan *Generator*

10. Selanjutnya mengeksplorasi dan menganalisis *dataset* pelatihan yang digunakan dalam tugas klasifikasi sampah.

```

base_dir = '/content/drive/MyDrive/Sampah Resnet/Train'
print(os.listdir(base_dir))

# Menghitung jumlah gambar pada dataset
number_label = {}
total_files = 0
for i in os.listdir(base_dir):
    counting = len(os.listdir(os.path.join(base_dir, i)))
    number_label[i] = counting
    total_files += counting

print("Total Files : " + str(total_files))

# Visualisasi jumlah gambar tiap kelas

plt.bar(number_label.keys(), number_label.values());
plt.title("Jumlah Gambar Tiap Label");
plt.xlabel('Label');
plt.ylabel('Jumlah Gambar');

```

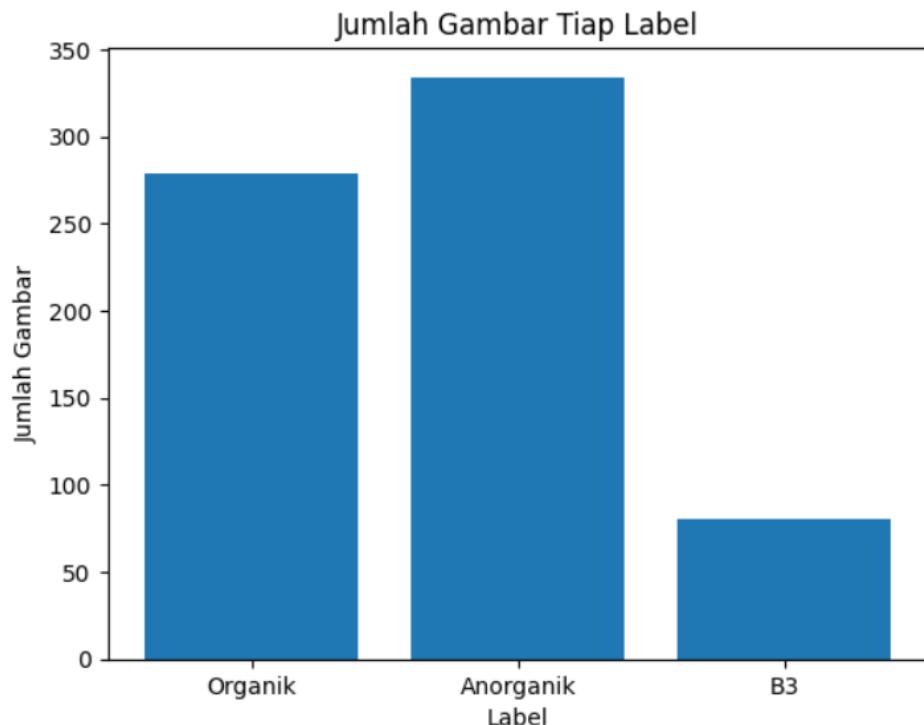
**Gambar 4.26** Eksplorasi dan Analisis Dataset

Outputnya menghasilkan kelas organik, anorganik, B3 dengan total gambar sebanyak 693 beserta grafik jumlah gambar tiap label.

```

['Organik', 'Anorganik', 'B3']
Total Files : 693

```



**Gambar 4.27** Grafik Jumlah Data per Kelas

11. Selanjutnya pada gambar 2.28 dan gambar 2.29 yaitu mendefinisikan generator data untuk proses pelatihan dengan *augmentasi*, serta membuat generator data untuk validasi tanpa *augmentasi*. *Outputnya* yaitu untuk pelatihan terdapat 3 kelas dengan total 693 gambar sampah dan untuk validasi terdapat 3 kelas dengan total 196 gambar sampah.

```
# Definisi generator data untuk pelatihan dengan augmentasi
datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    preprocessing_function=preprocess_input
)

# Contoh penggunaan generator untuk membaca dan augmentasi gambar dari direktori
train_generator_augmented = datagen.flow_from_directory(
    '/content/drive/MyDrive/Sampah_Resnet/Train',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    save_to_dir='/content/drive/MyDrive/Sampah_Resnet_Augmented', # Direktori penyimpanan augmented images
    save_prefix='aug',
    save_format='jpeg',
    shuffle=True
)
```

**Gambar 4.28 Augmentasi (1)**

```
# Banyaknya gambar yang akan digenerate (sesuai kebutuhan)
num_augmented_images = 1000

# Memulai proses augmentasi
for i in range(num_augmented_images // 32):
    batch = train_generator_augmented.next()

# Definisi generator data untuk validasi tanpa augmentasi
validation_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)

validation_generator = validation_datagen.flow_from_directory(
    '/content/drive/MyDrive/Sampah_Resnet/Valid',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False # Tidak perlu untuk diacak dalam data validasi
)
```

**Gambar 4.29 Augmentasi (2)**

12. Selanjutnya implementasi dari proses pelatihan model dengan menggunakan beberapa fungsi *callback* untuk memantau dan mengontrol pelatihan. Model disimpan dengan nama 'best\_model.h5'. Proses pelatihan model menggunakan data pelatihan dan validasi. *Callbacks* digunakan untuk menyimpan model terbaik berdasarkan akurasi pada data validasi (*Model Checkpoint*) dan menghentikan pelatihan jika tidak ada peningkatan akurasi (*Early Stopping*). *Callbacks* ini membantu mengoptimalkan pelatihan model dan mencegah *overfitting*.

```

checkpoint_filepath = 'best_model.h5'
model_checkpoint = ModelCheckpoint(
    checkpoint_filepath,
    save_best_only=True,
    monitor='val_accuracy', # Monitor val_accuracy untuk menyimpan model terbaik
    mode='max', # Mode max karena kita ingin memaksimalkan val_accuracy
    verbose=1
)

# Callback untuk menghentikan pelatihan jika tidak ada peningkatan val_accuracy
early_stopping = EarlyStopping(
    monitor='val_accuracy', # Monitor val_accuracy untuk early stopping
    patience=3, # Jumlah epoch tanpa peningkatan sebelum pelatihan dihentikan
    mode='max', # Mode max karena kita ingin memaksimalkan val_accuracy
    verbose=1
)

# Latih model
history = model.fit(
    train_generator,
    epochs=25,
    batch_size=32,
    callbacks=[model_checkpoint, early_stopping], # Callbacks digunakan selama pelatihan
    validation_data=validation_generator # Gunakan generator validasi untuk memonitor val_accuracy
)

```

**Gambar 4.30** Pelatihan Dataset

Dengan menggunakan *Early Stopping*, hasil pelatihan berhenti di 18 dari 25 kali pelatihan dikarenakan sudah mendapat akurasi yang terbaik.

```

Epoch 15/25
22/22 [=====] - ETA: 0s - loss: 0.1574 - accuracy: 0.9481
Epoch 15: val_accuracy improved from 0.96648 to 0.97207, saving model to best_model.h5
22/22 [=====] - 117s 5s/step - loss: 0.1574 - accuracy: 0.9481 - val_loss: 0.1333 - val_accuracy: 0.9721
Epoch 16/25
22/22 [=====] - ETA: 0s - loss: 0.1611 - accuracy: 0.9452
Epoch 16: val_accuracy did not improve from 0.97207
22/22 [=====] - 116s 5s/step - loss: 0.1611 - accuracy: 0.9452 - val_loss: 0.1260 - val_accuracy: 0.9609
Epoch 17/25
22/22 [=====] - ETA: 0s - loss: 0.1164 - accuracy: 0.9582
Epoch 17: val_accuracy did not improve from 0.97207
22/22 [=====] - 120s 6s/step - loss: 0.1164 - accuracy: 0.9582 - val_loss: 0.1233 - val_accuracy: 0.9553
Epoch 18/25
22/22 [=====] - ETA: 0s - loss: 0.1108 - accuracy: 0.9654
Epoch 18: val_accuracy did not improve from 0.97207
22/22 [=====] - 121s 6s/step - loss: 0.1108 - accuracy: 0.9654 - val_loss: 0.1264 - val_accuracy: 0.9553
Epoch 18: early stopping

```

**Gambar 4.31** Hasil Pelatihan

13. Selanjutnya untuk membuat dan menyimpan dua grafik: satu untuk melihat perkembangan akurasi (*Training* dan *Validation*) dan satu lagi untuk melihat perkembangan fungsi *loss* (*Training* dan *Validation*) selama pelatihan model.

```

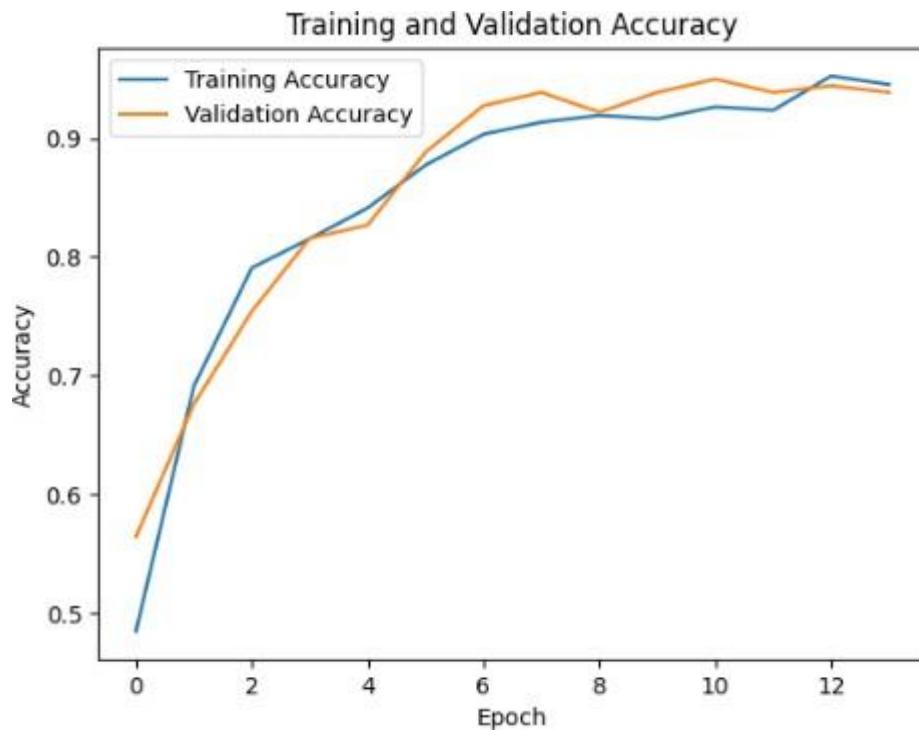
# Plot grafik akurasi pelatihan dan validasi
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.savefig('train_val_accuracy.png')
plt.show()

# Plot grafik loss pelatihan dan validasi
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.savefig('train_val_loss.png')
plt.show()

```

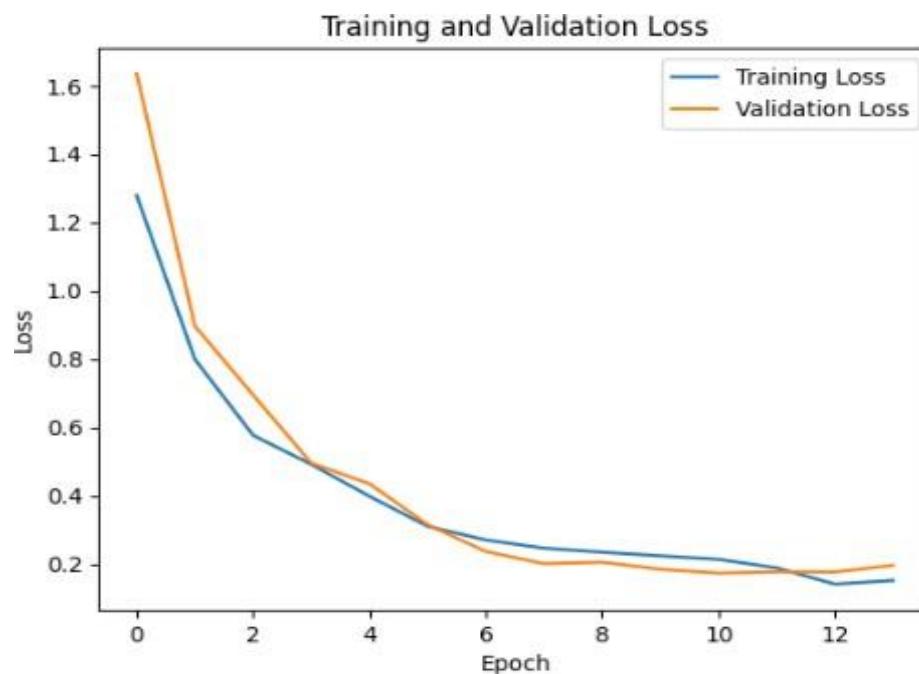
**Gambar 4.32** Membuat Grafik *Training* dan *Validation*

Pada gambar dibawah, terdapat grafik yang dihasilkan berdasarkan *accuracy* pelatihan dan validasi. Kedua garis terlihat saling berdekatan pada bagian atas grafik, menunjukkan bahwa model dapat melakukan prediksi dengan baik dari data pelatihan serta data yang baru pertama kali diperlihatkan.



**Gambar 4.33** Grafik Accuracy

Pada gambar di bawah grafik yang dihasilkan berdasarkan *loss* pelatihan dan validasi. Mengindikasikan bahwa model mempelajari pola data dengan baik pada saat pelatihan dan dapat mempelajari data baru dengan baik pula pada validasi.



**Gambar 4.34** Grafik Loss

14. Selanjutnya evaluasi model menggunakan metrik-metrik tertentu dan visualisasi hasil evaluasi, termasuk memuat kembali model terbaik dari *checkpoint*, melakukan prediksi pada data validasi, menghitung metrik evaluasi, dan menampilkan hasil metrik serta visualisasi *confusion matrix*.

```
# Memuat kembali model terbaik dari checkpoint
best_model = load_model('best_model.h5')

# Menggunakan generator validasi untuk memperoleh prediksi dan ground truth
validation_generator.reset()
y_pred_prob = best_model.predict(validation_generator)
y_pred = np.argmax(y_pred_prob, axis=1) # Menggunakan argmax untuk mendapatkan kelas prediksi
y_true = validation_generator.classes

# Menghitung metrik-metrik
accuracy = accuracy_score(y_true, y_pred)
conf_matrix = confusion_matrix(y_true, y_pred)
class_report = classification_report(y_true, y_pred)
roc_auc = roc_auc_score(y_true, y_pred_prob, multi_class='ovr')

# Menampilkan hasil metrik
print(f'Accuracy: {accuracy:.4f}')
print(f'Confusion Matrix:\n{conf_matrix}')
print('Classification Report:\n', class_report)
print(f'AUC-ROC: {roc_auc:.4f}')

# Visualisasi Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=validation_generator.class_indices.keys(),
            yticklabels=validation_generator.class_indices.keys())
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

**Gambar 4.35** Evaluasi dengan Metrik

Menghitung akurasi menggunakan ‘*accuracy\_score*’ dengan nilai yang didapat 0.9665.

```
6/6 [=====] - 28s 4s/step
Accuracy: 0.9721
```

**Gambar 4.36** Nilai Accuracy Model

*Confusion matrix* adalah alat evaluasi yang digunakan untuk menilai kinerja model klasifikasi terhadap suatu *dataset*. Terdiri dari empat sel yang mewakili *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), matriks ini memberikan gambaran tentang kemampuan model dalam membedakan antara kelas

positif dan negatif. Dengan menggunakan *confusion matrix*, kita dapat mengukur sejauh mana model mampu membedakan antara hasil yang benar dan salah dalam klasifikasi.

*Confusion matrix:*

$$\begin{bmatrix} 86 & 0 & 2 \\ 0 & 20 & 0 \\ 2 & 1 & 67 \end{bmatrix}$$

*Classification report* memberikan ringkasan statistik dari performa model klasifikasi pada setiap kelas. *Precision* mengukur seberapa banyak dari prediksi positif yang benar, *recall* mengukur seberapa banyak dari kelas positif yang berhasil dideteksi, dan *F1-score* adalah *harmonic mean* dari *precision* dan *recall*. *Support* adalah jumlah sampel yang termasuk dalam kelas tersebut.

*Classification report:*

**Tabel 4.3** Tabel Classification Report 1

No	Precision	Recall	F1 - Score	Support
0	0.98	0.97	0.97	89
1	0.91	1.00	0.95	20
2	0.97	0.96	0.96	70

**Tabel 4.4** Tabel Classification Report 2

	Precision	Recall	F1 - Score	Support
Accuracy			0.97	179
Macro Avg	0.95	0.97	0.96	179
Weighted Avg	0.97	0.97	0.97	179

15. Selanjutnya pada gambar 3.37, 3.38, 3.39 mendeklarasikan fungsi-fungsi *Python* untuk memuat dan memproses gambar, serta melakukan prediksi kategori sampah menggunakan model yang telah dilatih sebelumnya.

```

def load_and_preprocess_image(image_path):
    # Memuat gambar dari file
    img = image.load_img(image_path, target_size=(224, 224))

    # Mengonversi gambar menjadi larik numpy
    img_array = image.img_to_array(img)

    # Memperluas dimensi untuk membuat batch (diperlukan oleh beberapa model)
    img_array = np.expand_dims(img_array, axis=0)

    # Memproses gambar masukan
    img_array = preprocess_input(img_array)

    return img_array

```

**Gambar 4.37** Proses Prediksi Gambar (1)

```

def predict_waste_category(image_path, model):
    input_image = load_and_preprocess_image(image_path)

    # Prediksi menggunakan model
    prediction = model.predict(input_image)

    # Menampilkan skor prediksi mentah
    print("Skor Prediksi Mentah:", prediction)

    # Menentukan indeks kelas dengan probabilitas tertinggi
    predicted_class_index = tf.argmax(prediction, axis=1).numpy()[0]

    # Ambil indeks kelas dengan nilai prediksi tertinggi
    kelas_terprediksi = prediction.argmax()

    # Mendefinisikan kategori sampah
    waste_categories = {
        0: {"category": "Anorganik", "subcategories": ["Kertas/Tisu", "Plastik", "Styrofoam"]},
        1: {"category": "B3", "subcategories": ["Rokok"]},
        2: {"category": "Organik", "subcategories": ["Daun", "Kayu", "Sampah Buah"]}
    }

```

**Gambar 4.38** Proses Prediksi Gambar (2)

```

# Mendapatkan informasi kategori yang diprediksi
predicted_category_info = waste_categories[kelas_terprediksi]

return predicted_category_info

# Contoh penggunaan
model = tf.keras.models.load_model('/content/best_model.h5') # Gantilah 'path_to_your_model' dengan jalur yang sebenarnya
image_path = '/content/runs/detect/predict/20240107_091422.jpg.rf.d962b1d3e32db596d8fe48526cc8195b.jpg'
result = predict_waste_category(image_path, model)

print("Kategori Sampah:", result["category"])
print("Subkategori Sampah:", result["subcategories"])

```

**Gambar 4.39** Proses Prediksi Gambar (3)

Setelah memasukkan *path* dari *output* YOLO, hasilnya yaitu kategori sampah organik.

```
1/1 [=====] - 2s 2s/step
Skor Prediksi Mentah: [[ 0.039581 0.019101 0.94132]]
Kategori Sampah: Organik
Subkategori Sampah: ['Daun', 'Kayu', 'Sampah Buah']
```

**Gambar 4.40** Hasil Prediksi

#### 4.2.4 Pengujian model dari ResNet-50

Berikut ini adalah tahapan dalam proses pengujian model dari ResNet-50. Hasil prediksi didapat dari hasil percobaan model dengan gambar sampah sebanyak 85 gambar.

**Tabel 4.5** Pengujian Model ResNet-50

Nama Kelas	Hasil Prediksi Sebenarnya			
	Organik	Anorganik	B3	Tidak Terklasifikasi
Organik	33	3	1	
Anorganik	1	35	1	
B3	0	3	8	

$$\text{Accuracy} = \frac{\text{Total TP}}{\text{Total Data}}$$

$$\text{Accuracy} = \frac{33+35+8}{85} = \frac{76}{85} = 0.89$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

*Precision* untuk kelas “Organik”

$$TP = 33$$

$$FP = 1+0 = 1$$

*Precision* untuk kelas “Anorganik”

$$TP = 35$$

$$FP = 3+3 = 6$$

*Precision* untuk kelas “B3”

$$TP = 8$$

$$FP = 1+1 = 2$$

$$Precision = \frac{33+35+8}{34+41+10} = \frac{76}{85} = 0.89$$

$$Recall = \frac{TP}{TP+FN}$$

*Recall* untuk kelas “Organik”

$$TP = 33$$

$$FN = 3+1 = 4$$

*Recall* untuk kelas “Anorganik”

$$TP = 35$$

$$FN = 1+1 = 2$$

*Recall* untuk kelas “B3”

$$TP = 8$$

$$FN = 0+3 = 3$$

$$Recall = \frac{33+35+8}{37+37+11} = \frac{76}{85} = 0.89$$

$$F1 Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 Score = 2 \times \frac{0.89 \times 0.89}{0.89 + 0.89}$$

$$F1 Score = 2 \times \frac{0.7921}{1.78} = 0.89$$

### 4.3 Hasil Pengujian Sistem

Berikut merupakan hasil yang didapat dengan mendeteksi objek sampah dan kategori kelas sampah yang didapat kan melalu sistem yang telah dibuat.

**Tabel 4.6** Tabel Pengujian Sistem

No	Gambar Sampah	Hasil Deteksi	Hasil Klasifikasi	Keterangan
1		Sampah Buah	Organik	Benar
2		Plastik	Anorganik	Benar
3		Kertas atau Tisu	Anorganik	Benar
4		Rokok	B3	Benar
5		Daun	Organik	Benar
6		Styrofoam	Anorganik	Benar
7		Kayu	Organik	Benar

8		Daun		Salah
9		Kayu		Salah
10		Rokok	B3	Benar
11		Styrofoam	Anorganik	Benar
12		Daun	Organik	Benar
13		Kayu	Organik	Benar
14		Tidak Terdeteksi		Salah
15		Rokok	B3	Benar
16		Plastik	Anorganik	Benar

17		Sampah Buah	Organik	Benar
18		Kertas / Tisu	Anorganik	Benar
19		Daun	Organik	Benar
20		Plastik	Anorganik	Benar

Hasil pengujian dari 20 gambar sampah terdapat 17 hasil yang benar dan 3 salah. Kesalahan pada hasil deteksi disebabkan karena gambar yang dimasukkan pada *website* tidak terlihat jelas dan juga dikarenakan kurangnya variasi gambar pada *dataset* sehingga hasil deteksi tidak akurat. Sehingga didapat hasil akurasi jumlah benar/jumlah seluruhnya =  $17/20 = 85\%$

#### 4.4 Faktor Pengaruh Hasil Prediksi

Terdapat beberapa faktor yang mempengaruhi hasil deteksi dan klasifikasi sampah pesisir pantai antara lain:

1. Kualitas gambar

Kualitas pencahayaan, tidak *blur*, *angle* foto dan lain-lain. Semakin baik kualitas gambar, maka hasil deteksi semakin akurat.

2. Objek harus *full screen*

Usahakan foto sampah harus *full* layar dan jelas tidak terpotong, agar sistem dapat mendeteksi dengan tepat.

3. Tidak dapat memproses lebih dari 1 gambar

Untuk hasil yang akurat, gunakan 1 gambar sampah saja dalam melakukan prediksi, untuk 2 gambar juga bisa terdeteksi hanya saja terkadang hasilnya belum akurat.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan tahap implementasi yang telah dilakukan pada penelitian deteksi dan klasifikasi sampah pesisir Pantai Bunga Kabupaten Batubara menggunakan algoritma *You Only Look Once* (YOLO) v8 dan *Residual Network-50* dalam mengoptimalkan potensi sumber daya kelautan berkelanjutan yang diimplementasikan berbasis *website*, dapat disimpulkan bahwa:

1. Algoritma *You Only Look Once* (YOLO) versi 8 dapat digunakan untuk mendeteksi gambar sampah pesisir pantai.
2. Arsitektur *Residual Network- 50* dapat digunakan dalam proses klasifikasi sampah pesisir pantai ke dalam kelas organik, anorganik, dan B3.
3. Hasil pengujian model dari algoritma YOLO mendapat *accuracy* 0.87, *precision* 0.90, *recall* 0.89 dan *F1 Score* 0.89.
4. Nilai akurasi ResNet-50 pada proses pelatihan mendapat hasil 96.6%.
5. Hasil pengujian model dari metode ResNet-50 mendapat *accuracy* 0.89, *precision* 0.89, *recall* 0.89 dan *F1 Score* 0.89.
6. Hasil akhir pengujian sistem dalam mendeteksi gambar sampah mendapatkan akurasi 85%.
7. Hasil akhir dari sistem dapat menjadi acuan dan sumber informasi dalam mengetahui golongan jenis sampah untuk membantu menjaga kebersihan dan dapat membantu dalam mengoptimalkan potensi sumber daya kelautan berkelanjutan.

## 5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Disarankan untuk melakukan penelitian lebih lanjut dengan mengimplementasikan pada *Android* atau *iOS*.
2. Disarankan dalam membuat sistem yang dapat mengenali jenis sampah berbasis *real-time*.
3. Disarankan untuk melakukan penelitian lebih lanjut untuk menambahkan jumlah *dataset* yang lebih banyak dan lebih bervariasi.
4. Disarankan menggunakan algoritma YOLO versi 9 dan 10.
5. Disarankan untuk mencoba mengkombinasikan algoritma YOLO dengan metode *Efficient Net*.
6. Disarankan menggunakan algoritma lain seperti *Faster R-CNN*, *Single Shot MultiBox Detector* (SSD), dan *R-CNN* sebagai perbandingan akurasi.

## DAFTAR PUSTAKA

- Amwin, A. (2021). Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO). *Universitas Islam Indonesia*, 1–60. <https://dspace.uii.ac.id/handle/123456789/34154>
- Ayunda, R., Tantimin, Febriyani, E., Agustini, S., Patros, A., & Nurisman, E. (2023). Gerakan Bebas Sampah di Pulau Mumbut Laut Upaya Perlindungan Hukum Potensi Kelautan. *Sang Sewagati Journal*, vol 1, 42.
- Cahyadi, F. D., Widiyanto, K., & Prakoso, K. (2021). EDUKASI GERAKAN BERSIH PANTAI DAN LAUT DI PULAU TUNDA. *ABDIMAS UNWAHAS*, 6(1). <https://doi.org/10.31942/abd.v6i1.4432>
- Djongihi, O. A., Adjum, S., Salam, R., Program, D., Geografi, S., & Khairun, U. (n.d.). *DAMPAK PEMBUANGAN SAMPAH DI PESISIR PANTAI TERHADAP LINGKUNGAN SEKITAR (STUDI KASUS MASYARAKAT PAYAHE KECAMATAN OBA KOTA TIDORE KEPULAUAN)*.
- Harefa, M. S., Salsabila, G., Syahputra, I., & Salsabila<sup>4</sup>, V. A. (2022). *MUDABBIR (Journal Research and Education Studies)* (Vol. 2, Issue 2).
- Herdiansyah, H., Saiya, H. G., Afkarina, K. I. I., & Indra, T. L. (2021). Coastal community perspective, waste density, and spatial area toward sustainable waste management (Case study: Ambon bay, Indonesia). *Sustainability (Switzerland)*, 13(19). <https://doi.org/10.3390/su131910947>
- Husain, R., & Saleh, M. (2022). Pengelolaan Lingkungan Pesisir Melalui Gerakan Bersih Pantai Dan Pemanfaatan Barang Bekas Sebagai Upaya Mengurangi Sampah Di Desa Biluhu Timur Kabupaten Gorontalo. *Dikmas: Jurnal Pendidikan Masyarakat Dan Pengabdian*, 2(1), 191. <https://doi.org/10.37905/dikmas.2.1.191-202.2022>
- Junzhe, Z., Fuqiang, J., Yupeng, C., Weiyi, W., & Qing, W. (2023). A water surface garbage recognition method based on transfer learning and image enhancement. *Results in Engineering*, 19. <https://doi.org/10.1016/j.rineng.2023.101340>
- Lin, F., Hou, T., Jin, Q., & You, A. (2021). Improved Yolo based detection algorithm for floating debris in waterway. *Entropy*, 23(9). <https://doi.org/10.3390/e23091111>
- Md, Z., Amin, A., Sami, N., & Hassan, R. (2021). An Approach of Classifying Waste Using Transfer Learning Method. *International Journal on Perceptive and Cognitive Computing (IJPCC)*, 7(1), 41.

- Nufus, N., Ariffin, D. M., Satyawan, A. S., Nugraha, R. A. S., Asysyakuur, M. I., Marlina, N. N. A., ... & Ema, E. (2021, December). Sistem Pendekripsi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi. In Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO) (Vol. 3, pp. 123-134).
- Reynaldi Tanjung, K., & Juwianto, H. (2022). Klasifikasi Benda Organik dan Anorganik Dengan Metode YOLOv3 dan ResNet50.
- Septiani, S., & Susilawati, S. (2023). Upaya Pemberdayaan Masyarakat. *Jurnal Ventilator: Jurnal Riset Ilmu Kesehatan Dan Keperawatan*, Vol. 1, 1–148.
- Syarifah, R. D., Amini, H. W., Nihayah, H., & Luthfiyana, N. U. (2022). TRASH CAN-COMPOSTER: ALAT PENCACAH SAMPAH ORGANIK UNTUK PENCACAH SAMPAH LIMBAH PERTANIAN. *JMM (Jurnal Masyarakat Mandiri)*, 6(3), 1712. <https://doi.org/10.31764/jmm.v6i3.7668>
- Terven, J., & Cordova-Esparza, D. (2023). *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. <https://doi.org/10.3390/make5040083>
- Wahyutama, A. B., & Hwang, M. (2022). YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins. *Electronics (Switzerland)*, 11(9). <https://doi.org/10.3390/electronics11091323>