Penggunaan Algoritma YOLOv8 untuk Deteksi Jenis Sampah: Studi Implementasi di Website Bank Sampah Bersinar

1st Rifqi Fadhila Shandi
School of Electrical Engineering
Telkom University
Bandung, Indonesia
rifqishandi@student.telkomuniversity.ac.id

2nd Meta Kallista
School of Electrical Engineering
Telkom University
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

3rd Casi Setianingsih
School of Electrical Engineering
Telkom University
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

***Abstrak* — Di Indonesia, pengelolaan sampah yang efektif merupakan tantangan besar, terutama mengingat jumlah sampah yang dihasilkan terus meningkat setiap tahunnya. Tujuan dari penelitian ini adalah untuk meningkatkan akurasi dan efisiensi deteksi dan pemilahan sampah menggunakan model deep learning YOLOv8, yang dikembangkan untuk deteksi objek secara real-time. Platform Roboflow digunakan untuk memproses dan menganalisis dataset gambar sampah yang dikumpulkan dari sumber online seperti Google dan Kaggle. Model YOLOv8 yang dilatih diterapkan dalam dua skenario utama: deteksi objek secara langsung melalui webcam dan implementasi pada antarmuka website. Hasil pengujian menunjukkan bahwa YOLOv8 memiliki kemampuan mendeteksi berbagai jenis sampah dengan sangat akurat; pada konfigurasi tertentu, nilai mAP mencapai 0,91490. Hasilnya menunjukkan bahwa teknologi ini dapat membantu meningkatkan pengelolaan sampah di Indonesia, terutama di Bank Sampah Bersinar dan sistem pengelolaan sampah lainnya.**

Kata kunci— YOLOv8, deteksi objek, deep learning, pengolahan sampah

I. PENDAHULUAN

Seiring dengan bertambahnya populasi, Indonesia kini dihadapkan pada tantangan besar dalam pengelolaan sampah. Pertumbuhan penduduk yang pesat dan minimnya infrastruktur pengolahan sampah yang memadai membuat lonjakan volume sampah menjadi masalah mendesak yang harus segera diatasi[1]. Berdasarkan data dari Sistem Informasi Pengelolaan Sampah Nasional (SIPSN), pada tahun ini saja, Indonesia menghasilkan 36 juta ton sampah, namun hanya sekitar 62% yang berhasil dikelola dengan baik. Situasi ini menunjukkan bahwa sistem pengelolaan sampah negara harus diperbaiki segera[2].

Hambatan utama untuk pengolahan sampah yang efektif adalah kurangnya kesadaran masyarakat tentang pentingnya pengelolaan sampah, kurangnya pemahaman tentang manfaat daur ulang, dan keterbatasan ekonomi. Pengelolaan sampah yang efektif mencakup pendekatan komprehensif, mulai dari pengurangan produksi hingga daur ulang melalui sistem pengumpulan dan pemilahan yang efektif, sering terhambat oleh infrastruktur, kepadatan penduduk, dan kondisi sosial ekonomi.

Dalam upaya mengatasi masalah pengelolaan sampah yang belum optimal, terdapat berbagai tantangan kompleks yang saling terkait, salah satunya adalah identifikasi dan pengelolaan berbagai jenis sampah. Berdasarkan hasil survei di salah satu bank sampah di Bandung menunjukkan adanya berbagai kategori sampah, termasuk organik, non-organik, dan elektronik. Oleh karena itu, pengenalan dan pemilahan yang akurat dari jenis-jenis sampah ini menjadi fokus utama dalam pengembangan aplikasi ini. Untuk mengatasi tantangan ini, diperlukan pendekatan komprehensif yang melibatkan teknologi mutakhir seperti deep learning. YOLOv8, sebuah sistem deteksi objek berbasis deep learning, menawarkan potensi besar untuk meningkatkan efisiensi dalam identifikasi dan pemilahan sampah. Dengan pemanfaatan teknologi ini, diharapkan dapat mendukung operasional bank sampah dan memperbaiki sistem pengelolaan sampah di Indonesia secara keseluruhan.

II. KAJIAN TEORI

A. Deep learning

Deep learning (DL) adalah cabang ilmu dari machine learning (ML) dan artificial intelligence (AI) yang telah menjadi teknologi penting dalam Revolusi Industri 4.0. Berbasis pada Artificial Neural Network (ANN), DL memiliki kemampuan untuk belajar dari data secara mandiri dan telah menjadi topik penting di berbagai bidang komputasi. DL merepresentasikan data secara hierarkis dengan menggunakan berbagai lapisan dan menghasilkan model komputasi yang mampu mengenali pola – pola yang kompleks. Aplikasi seperti pengenalan visual, analitik teks, dan keamanan siber telah menggunakan hasil model Deep Learning, meskipun pelatihannya memerlukan waktu yang lama karena banyaknya parameter yang harus dipelajari[3].

B. YOLO

YOLO (You Only Look Once) adalah algoritma deteksi objek yang pertama kali dirilis pada tahun 2016. Algoritma ini mampu melakukan deteksi objek secara real-time dengan kecepatan yang sangat tinggi, mencapai hingga 45 frame per detik (FPS) dalam pengujian dengan konfigurasi yang sama, menjadikannya inovasi dalam deteksi objek. YOLO mengubah seluruh proses deteksi objek menjadi masalah regresi, di mana lokasi dan kategori objek hanya dapat diperoleh dengan menggunakan satu jaringan saraf dan seluruh gambar digunakan sebagai input[4].

Dengan membagi gambar menjadi grid, YOLO menggunakan pendekatan unik untuk memprediksi bounding boxes dan kemungkinan keberadaan objek di setiap grid. Tidak seperti pendekatan *two-stage object detection* yang lebih kompleks, YOLO menggabungkan deteksi dan klasifikasi dalam satu langkah, yang membuat proses lebih mudah. YOLO sangat cocok untuk aplikasi yang membutuhkan deteksi objek cepat, seperti robotika, sistem pengawasan, dan kendaraan otonom.

C. TensorFlow.js

TensorFlow.js (tfjs) merupakan sebuah library open-source yang dibangun berdasarkan TensorFlow versi Python, menawarkan API yang sebanding dengan TensorFlow versi Python untuk pengembangan dan pelatihan model machine learning (ML) di ekosistem JavaScript. Dengan tfjs, pengembang dapat membangun, melatih, dan menjalankan model ML secara lokal atau di cloud, serta menggunakan model ML yang sudah ada untuk aplikasi berbasis web[5].

D. Roboflow

Roboflow adalah platform komprehensif yang mempermudah pengembangan proyek computer vision dengan menyediakan berbagai alat dan layanan untuk setiap tahap pengembangan model. Platform memungkinkan pengguna mengimpor, menganotasi, dan mengelola dataset gambar, serta menyediakan alat untuk augmentasi data untuk meningkatkan variasi dan kualitas dataset. Roboflow juga mendukung ekspor dataset ke berbagai format, dan itu kompatibel dengan framework machine learning populer seperti TensorFlow, PyTorch, dan YOLO[6].

E. Mean Average Precision (mAP)

mAP merupakan metriks evaluasi yang umum digunakan dalam bidang pengelolaan citra dan deteksi objek, khusunya di dalam konteks tugas – tugas yang berkaitan dengan deteksi objek dalam gambar. Dalam konteks untuk objek deteksi, mAP mengukur seberapa baik suatu model dapat melakukan identifikasi dan menepatkan objek pada lokasi yang tepat dalam gambar. Pada metriks ini menggabungkan dua konsep utana yaitu precision dan recal, dimana mAP menggabungkan presisi dan recall pada berbagai nilai ambang batas (threshold) untuk menghasilkan skor rata-rata. Semakin tinggi mAP, semakin baik kinerja model deteksi objeknya[7].

III. METODE

A. Persiapan Dataset

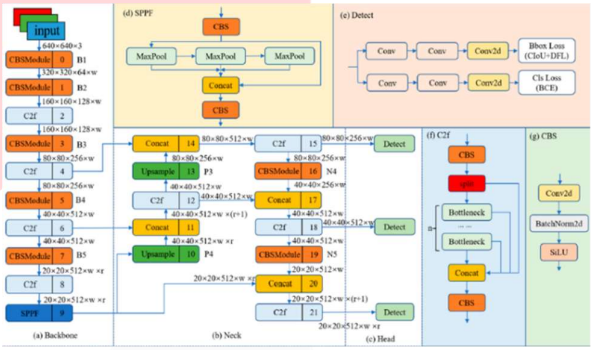
Tahapan penting dalam melatih model YOLOv8 untuk mendeteksi objek, terutama jenis sampah, adalah persiapan dataset. Dimulai dengan pengumpulan gambar yang menunjukkan jenis sampah yang ada di Bank Sampah Bersinar. Gambar-gambar ini dikumpulkan dari berbagai sumber online, termasuk Google dan dataset publik Kaggle. Setiap gambar kemudian dianotasi untuk menandai objek yang ingin dideteksi, dan label diberikan untuk mengklasifikasikan objek tersebut.

Dataset kemudian dibagi menjadi beberapa bagian, termasuk data pelatihan, validasi, dan uji, untuk memastikan bahwa model dapat dilatih dan diuji dengan benar. Langkah berikutnya adalah pre-processing, di mana gambar-gambar dalam dataset disesuaikan untuk memiliki ukuran dan orientasi yang sama. Terakhir, augmentasi dilakukan untuk meningkatkan variasi data sehingga model dapat belajar dari berbagai kondisi dengan lebih baik.

Metode ini memastikan bahwa dataset yang digunakan memiliki kualitas tinggi dan siap untuk digunakan dalam melatih model YOLOv8 untuk meningkatkan akurasi dan efisiensi deteksi objek.

B. YOLOv8

YOLOv8 diperkenalkan oleh Ultralytics pada 10 Januari 2023. Berdasarkan berbagai pengujian yang dilakukan YOLOv8 memberikan akurasi dan kecepatan deteksi yang lebih tinggi dibandingkan dengan para seri sebelumnya seperti YOLOv5 dan YOLOv7.



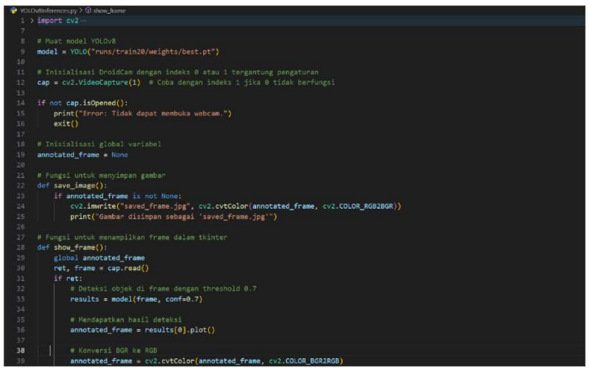
Gambar 1 Arsitektur YOLOv8

Berdasarkan Gambar 1 YOLOv8 memiliki tiga bagian utama dalam arsitekturnya yaitu backbone, neck, dan head, seperti yang ditunjukkan pada Gambar 1

1. Backbone : Backbone YOLOv8 menggunakan struktur hirarkis untuk ekstraksi fitur dari gambar masukan.
2. Neck : Neck bertugas menggabungkan dan memproses ulang fitur dari Backbone untuk meningkatkan kualitas deteksi objek pada berbagai skala.
3. Head : Head bertugas melakukan prediksi bounding box, klasifikasi objek, dan deteksi pada berbagai skala.

C. Penerapan YOLOv8 untuk Deteksi Objek Langsung Menggunakan Webcam

Pada Gambar 2 source code digunakan untuk menampilkan output dari webcam secara real-time dengan bantuan Library OpenCV menggunakan bahasa pemrograman Python. Dengan menggunakan source code tersebut , output dari webcam dapat dipanggil menggunakan Python dan dikonfigurasi dengan algoritma YOLOv8 untuk mendeteksi objek secara langsung.



```python
import cv2

# Muat model YOLOv8
model = YOLO("runs/train20/weights/best.pt")

# Inisialisasi DroidCam dengan indeks 0 atau 1 tergantung pengaturan
cap = cv2.VideoCapture(1)  # Coba dengan indeks 1 jika 0 tidak berfungsi

if not cap.isOpened():
    print("Error: Tidak dapat membuka webcam.")
    exit()

# Inisialisasi global variabel
annotated_frame = None

# Fungsi untuk menyimpan gambar
def save_image():
    if annotated_frame is not None:
        cv2.imwrite("saved_frame.jpg", cv2.cvtColor(annotated_frame, cv2.COLOR_RGB2BGR))
        print("Gambar disimpan sebagai 'saved_frame.jpg'")

# Fungsi untuk menampilkan frame dalam tkinter
def show_frame():
    global annotated_frame
    ret, frame = cap.read()
    if ret:
        # Deteksi objek di frame dengan threshold 0.7
        results = model(frame, conf=0.7)

        # Mendapatkan hasil deteksi
        annotated_frame = results[0].plot()

        # Konversi BGR ke RGB
        annotated_frame = cv2.cvtColor(annotated_frame, cv2.COLOR_BGR2RGB)
```

Gambar 2 Penerapan YOLOv8 untuk Deteksi Objek Langsung Menggunakan Webcam

D. Penerapan TensorFlow.js pada Algoritma YOLOv8

Pada Gambar 3 merupakan source code untuk memuat model YOLOv8 dan mengekspornya ke dalam format TensorFlow.js menggunakan bahasa Python. Dengan menggunakan source code di atas, model YOLOv8 yang telah dilatih dapat diubah ke format yang kompatibel dengan TensorFlow.js, memungkinkan penggunaannya di aplikasi berbasis web.

```python
from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n.pt")  # load an official model
model = YOLO("runs\\detect\\train54\\weights\\best.pt")  # load a custom trained model

# Export the model
model.export(format="tfjs")
```

Gambar 3 Penerapan TensorFlow.js pada Algoritma YOLOv8

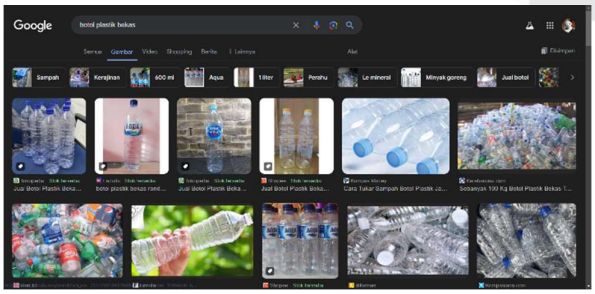
IV. HASIL DAN PEMBAHASAN

A. Persiapan Dataset

Pada pengujian persiapan dataset dibagi kedalam enam tahapan yaitu :

1. Pengumpulan Dataset

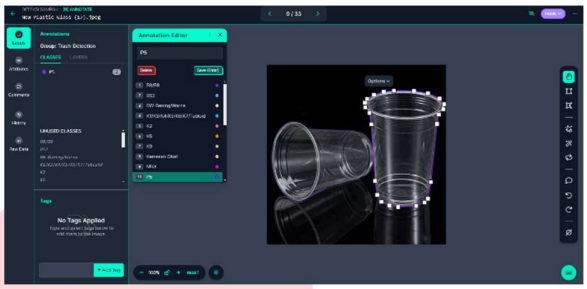
Dataset yang dikumpulkan berupa gambar, proses pengumpulan dataset adalah melalui mesin pencarian Google serta dataset publik yang tersedia di Kaggle, dengan fokus pada data jenis sampah dari Bank Sampah Bersinar. Setelah semua gambar terkumpul, dataset diunggah ke platform Roboflow. Tujuan pengumpulan dataset ini adalah untuk melatih model pembelajaran mesin untuk melakukan tugas tertentu, seperti deteksi objek, klasifikasi gambar, atau segmentasi gambar. Model yang dibuat dari dataset yang berkualitas tinggi akan lebih akurat dan efisien dalam menemukan pola atau fitur yang diinginkan dari gambar.



Gambar 4 Pengumpulan Dataset

2. Anotasi Dataset

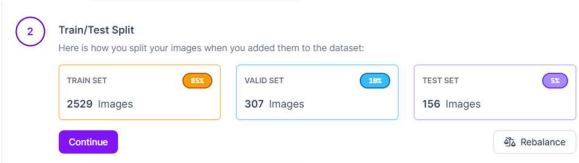
Setelah proses unggah seluruh gambar yang akan dijadikan bahan untuk dataset, selanjutnya adalah proses anotasi objek pada gambar yang telah diunggah sebelumnya. Setiap gambar dianotasi menggunakan bounding box, proses ini dilakukan untuk menandai dan mengidentifikasi area spesifik dari objek sampah yang ingin dideteksi. Bounding box memberikan koordinat yang mendefinisikan lokasi dan ukuran objek dalam gambar. Setelah pemberian bounding box, proses dilanjutkan dengan memberikan labeling untuk mengklasifikasikan jenis – jenis sampah yang ada di dalam gambar.



Gambar 5 Anotasi Objek

3. Pembagian Dataset

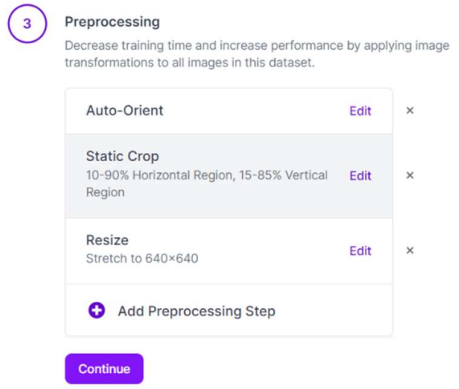
Setelah seluruh gambar selesai di anotasikan, maka langkah selanjutnya adalah melakukan pembagian dataset, dimana dataset terdapatd dua jenis pembagian dataset yang pertama dataset akan dibagi menjadi 3 bagian, yaitu Data Pelatihan (Train Set), Data Validasi (Validation Set), dan Data Uji (Test Set). Presentase pembagian dataset yang dibuat adalah 85% untuk Train Set, 10% untuk Validation Set, dan 5% untuk Test Set.



Gambar 6 Pembagian Dataset

4. Pre-processing Dataset

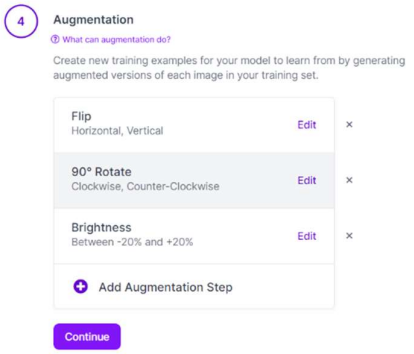
Pada bagian ini setelah dataset dibagi menjadi tiga bagian, langkah selanjutnya adalah melakukan pre-processing pada dataset. Langkah ini bertujuan untuk menyeimbangkan dataset agar memiliki ketentuan yang seragam. Dengan dataset yang seimbang, pelatihan algoritma YOLO dapat berlangsung secara konsisten, yang pada gilirannya mempercepat proses pelatihan algoritma YOLO dalam mengenali objek sampah yang sudah diberi bounding box dan label. Pada tahap ini, kami melakukan Auto-Orient, Static Crop, dan Resize gambar pada dataset. Hal ini dilakukan agar semua gambar memiliki ukuran yang sama, sehingga dataset menjadi seimbang.



Gambar 7 Pre-processing Dataset

5. Augmentasi Dataset

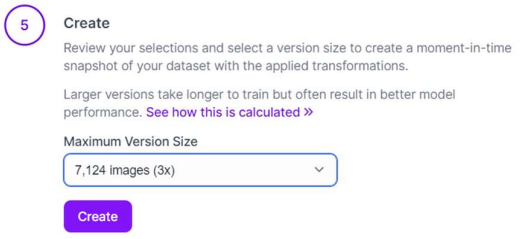
Selanjutnya adalah masuk pada tahap augmentasi, tujuan dari tahap ini adalah untuk memperbanyak jumlah dataset dengan cara memodifikasi citra pada gambar dataset. Augementasi yang dilakukan pada dataset meliputi melakukan flip pada gambar, rotate pada gambar dan mengantur tingkat kecerahan pada gambar.



Gambar 8 Augmentasi Dataset

6. Generate Dataset

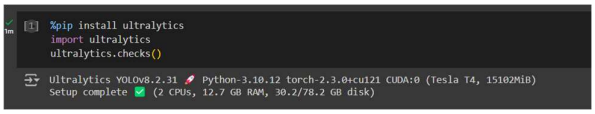
Langkah ini merupakan tahap terakhir dalam pembuatan dataset. Tujuan dari meng-generate dataset adalah untuk membatasi duplikasi pada dataset asli. Pada langkah augmentasi dan pre-processing dataset, hasil dari kedua proses tersebut adalah duplikasi yang telah dimodifikasi dari dataset asli. Oleh karena itu, pada tahap ini ditetapkan batasan untuk duplikasi tersebut agar dataset dapat digandakan sesuai dengan kebutuhan.



Gambar 9 Generate Dataset

B. Konfigurasi Algoritma YOLOv8

Dalam pembuatan deteksi objek berupa jenis sampah algoritma yang digunakan adalah YOLOv8. Algoritma ini dibuat oleh Ultralytics yang meruapakan pengembang dari algoritma YOLO dari versi awal hingga yang terbaru saat ini. Kloning repository algoritma YOLOv8 dari Github digunakan untuk membantu dalam proses pembuatan fitur deteksi objek sampah.



Gambar 10 Konfigurasi Algoritma YOLOv8

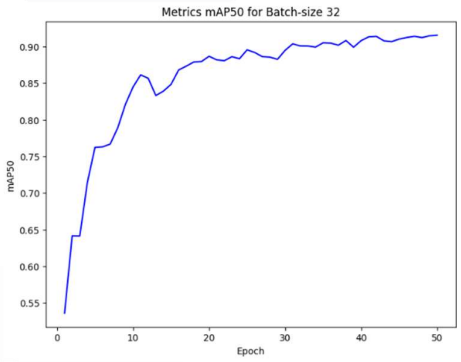
C. Training Model

Selanjutnya adalah tahapan proses training model untuk algoritma. Hal ini bertujuan agar algoritma dapat melatih dirinya untuk mengenali objek sesuai dengan dataset yang akan dilatih. Untuk mendapatkan parameter terbaik ,dilakukan beberapa pengujian yang dilakukan terdapat empat pengujian yang dilakukan, yaitu :

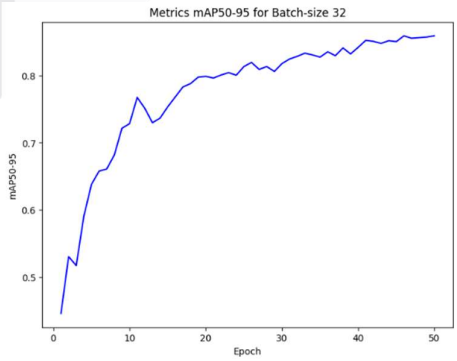
1. Pengujian Batch-size

Tabel 1 Pengujian Batch-size

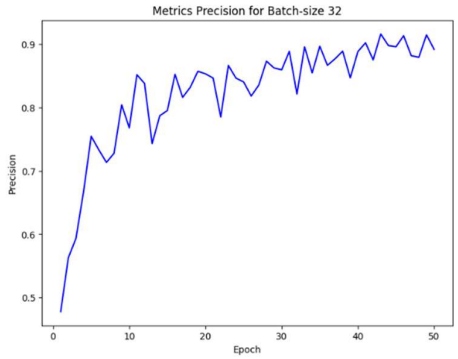
Batch-size	mAP_0.5	mAP_0.5:0.95	Precision	Recall
16	0.89978	0.82135	0.85456	0.83998
32	0.91549	0.859	0.89148	0.86352
64	0.91312	0.86096	0.90808	0.84891



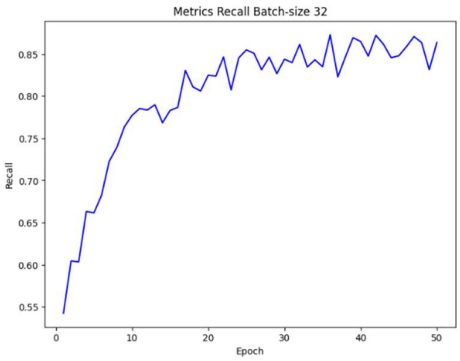
Gambar 11 mAP50 Batch-size 32



Gambar 12 mAP50-95 Batch-size 32



Gambar 13 Precision Batch-size 32



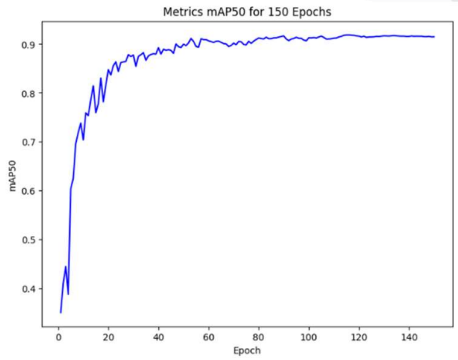
Gambar 14 Recal Batch-size 32

Berdasarkan data hasil pengujian diatas secara keseluruhan, Batch-size 32 memiliki kinerja yang paling konsisten dan terbaik di sebagian besar metrik, dengan nilai tertinggi kedua di mAP_0.5:0.95 dan nilai tertinggi pertama di mAP_0.5 serta Recall. Meskipun Batch-size 64 memiliki Precision tertinggi, nilai Recall-nya sedikit lebih rendah dibandingkan dengan Batch-size 32.

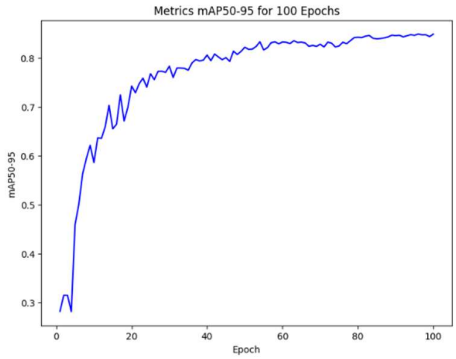
2. Pengujian Epoch

Tabel 2 Pengujian Epoch

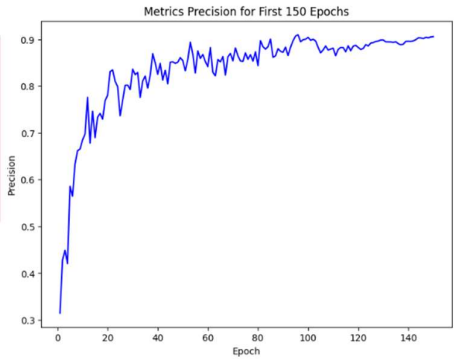
Epoch	mAP_0.5	mAP_0.5:0.95	Precision	Recall
50	0.89978	0.82135	0.85456	0.83998
100	0.91281	0.84822	0.90588	0.83687
150	0.91490	0.85986	0.90588	0.85989



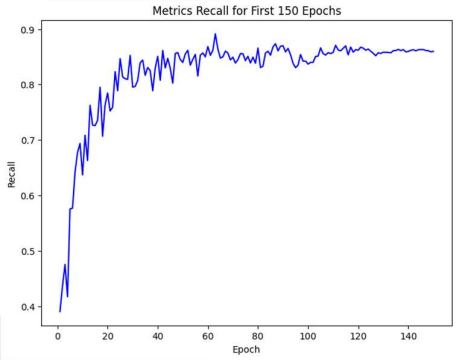
Gambar 15 mAP50 Epoch 150



Gambar 16 mAP50-95 Epoch 150



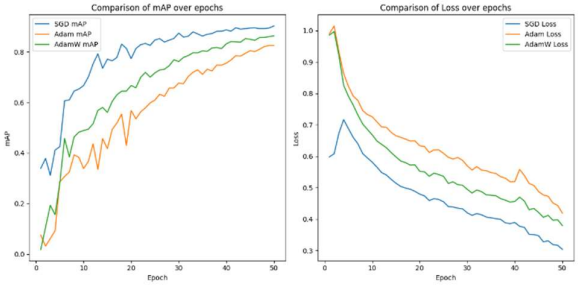
Gambar 17 Precision Epoch 150



Gambar 18 Recall Epoch 150

Berdasarkan dari pengujian 3 Epochs, dapat disimpulkan bahwa Epoch terbaik untuk hasil training pendeteksian yang akan digunakan adalah Epoch = 150. Dimana nilai dari Epoch 150 ini memiliki nilai mAP_0.5 0.91490, mAP_0.5:0.95 0.85986, Precision 0.90588, dan Recall 0.85989.

3. Pengujian Optimizer



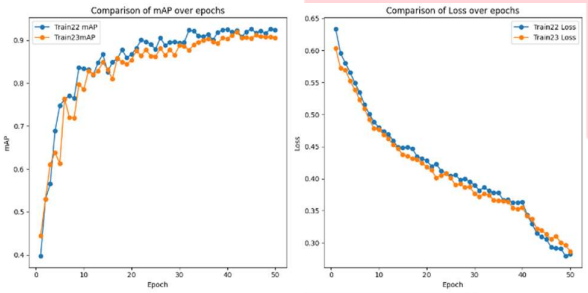
Gambar 19 Grafik Perbandingan Pengujian Optimizer

```
Best mAP for SGD: 0.9020
Best mAP for Adam: 0.8250
Best mAP for AdamW: 0.8626
```

Gambar 20 Hasil Perbandingan Pengujian Optimizer

Berdasarkan perbandingan hasil dari ketiga jenis optimizer yang telah diujikan, diketahui bahwa optimizer SGD memberikan hasil mAP yang lebih baik dibandingkan dengan kedua optimizer lainnya, dengan nilai mAP sebesar 0.9020. Oleh karena itu, optimizer yang akan digunakan dalam fitur trash detection yang akan dibuat adalah menggunakan optimizer SGD.

4. Pengujian Split Dataset



Gambar 21 Grafik Perbandingan Pengujian Split Dataset

```
Best mAP for Train22: 0.9020
Best mAP for Train23: 0.8250
```

Berdasarkan dari hasil training menggunakan dua jenis split dataset yang berbeda, di dapatkan bahwa split data dengan pembagian 85% Training Set, 10% Validation Set, 5 Test Set memberikan hasil mAP yang jauh lebih baik yaitu sebesar 0.9020. Sehingga pada proses training model nantinya akan digunakan split dataset 85% Training Set, 10% Validation Set, 5 Test Set.

D. Pengujian Model YOLOv8 Secara Real-Time

Pengujian model YOLOv8 secara real-time, baik menggunakan webcam maupun pada platform website. Tujuan dari pengujian ini adalah untuk menilai kinerja dan akurasi model dalam mendeteksi objek secara langsung dari video streaming yang ditangkap oleh webcam, serta implementasi model pada antarmuka website.

1. Hasil Pengujian Penerapan YOLOv8 Menggunakan Webcam Secara Real-Time

Tabel 3 Hasil Pengujian secara real-time

2. Hasil Implemetasi Model YOLOv8 Pada Antarmuka Website

Gambar 22 Input Gambar Deteksi ada Antarmuka Website

Gambar 23 Output Hasil Deteksi pada Antarmuka Website

V. KESIMPULAN

Implementasi algoritma YOLOv8 dalam penelitian ini menunjukkan hasil yang memuaskan dalam konteks deteksi jenis sampah secara real-time. Model YOLOv8 memiliki akurasi deteksi yang tinggi, seperti yang ditunjukkan oleh nilai mAP yang optimal pada konfigurasi tertentu. Ini dicapai dengan menggunakan dataset yang

dikumpulkan dari berbagai sumber online dan diproses melalui platform Roboflow. Pengujian menunjukkan bahwa teknologi ini dapat digunakan untuk meningkatkan efisiensi pengelolaan sampah, terutama dalam proses pemilahan yang lebih cepat dan tepat. Di masa depan, keberhasilan ini akan memungkinkan penerapan YOLOv8 pada skala yang lebih besar dan integrasinya dengan sistem pengelolaan sampah yang lebih kompleks.

REFERENSI

[1] J. M. Kadang and N. Sinaga, "Pengembangan Teknologi Konversi Sampah Untuk Efektifitas Pengolahan Sampah dan Energi Berkelanjutan," *TEKNIKA: Jurnal Ilmiah Bidang Ilmu Rekayasa*, vol. 15, no. 1, pp. 33–44, 2021.

[2] SIPSN, "Kementrian Lingkungan Hidup dan Kehutanan, "Sistem Informasi Pengelolaan Sampah Nasional (SIPSN)."

[3] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2 (6): 420," 2021.

[4] X. Cong, S. Li, F. Chen, C. Liu, and Y. Meng, "A review of YOLO object detection algorithms based on deep learning," *Frontiers in Computing and Intelligent Systems*, vol. 4, no. 2, pp. 17–20, 2023.

[5] Ultralytics, "https://docs.ultralytics.com/integrations/tfjs/#why-should-you-export-to-tfjs."

[6] Ultralytics, "https://docs.ultralytics.com/integrations/roboflow/. "

[7] Z. Karimi, "Confusion Matrix." [Online]. Available: https://www.researchgate.net/publication/355096788