JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

DETEKSI SAMPAH ORGANIK DAN ANORGANIK MENGGUNAKAN MODEL YOLOV8

Bima Prasetio[1)], Nunik Pratiwi*[2)]
1. Universitas Muhammadiyah Prof. Dr. Hamka, Indonesia
2. Universitas Muhammadiyah Prof. Dr. Hamka, Indonesia

Article Info
Kata Kunci: *Intersection over Union* (*IoU*); *mean Average Precision* (*mAP*); Sampah Organik dan Anorganik; YOLOv8

Keywords: *Intersection over Union (IoU); mean Average Precision (mAP); Organic and Anorganic Waste; YOLOv8*

Article history:
Received 14 October 2024
Revised 21 November 2024
Accepted 20 December 2024
Available online 1 March 2025

DOI :
https://doi.org/10.29100/jipi.v10i1.5965

* Corresponding author.
Corresponding Author
E-mail address:
npratiwi@uhamka.ac.id

ABSTRAK
Penelitian ini bertujuan untuk mengembangkan model deteksi sampah organik dan anorganik menggunakan algoritma YOLOv8 dengan penambahan noise pada citra seperti normal, *gaussian blur*, *darkness*, *low resolution*, dan *motion blur*. Deteksi sampah yang efektif sangat penting untuk pengelolaan limbah yang berkelanjutan. Dataset yang digunakan terdiri dari 1178 citra yang dikumpulkan dari Roboflow dan dibagi menjadi set pelatihan, validasi, dan pengujian. Model YOLOv8 dilatih menggunakan teknik *transfer learning* selama 30 epoch dengan ukuran gambar 640x640 piksel. Evaluasi dilakukan terhadap metrik performa seperti *precision*, *recall*, *F1-Score*, *mAP*, dan *IoU* dalam berbagai kondisi noise. Hasil penelitian menunjukkan bahwa model mencapai performa terbaik pada kondisi normal dengan *precision* sebesar 0,98, *recall* 1,00, *F1-Score* 0,99, dan *IoU* 0,84. Pada kondisi *gaussian blur*, model menunjukkan *precision* 1,00 namun sedikit penurunan pada *recall* menjadi 0,93 dan *F1-Score* 0,96. Kondisi *darkness* dan *low resolution* menunjukkan sedikit penurunan metrik dengan *F1-Score* masing-masing sebesar 0,98. Kondisi *motion blur* adalah yang paling menantang dengan penurunan signifikan pada *precision* menjadi 0,96, *recall* 0,96, *F1-Score* 0,96, dan *IoU* 0,80.

ABSTRACT
This research aims to develop an organic and inorganic waste detection model using the YOLOv8 algorithm with the addition of noise in the image such as nor-mal, gaussian blur, darkness, low resolution, and motion blur. Effective litter detection is essential for sustainable waste management. The dataset used consists of 1178 images collected from Ro-boflow and divided into training, validation, and testing sets. The YOLOv8 model was trained using transfer learning techniques for 30 epochs with an image size of 640x640 pixels. Evaluation was conducted on performance metrics such as precision, recall, F1-Score, mAP, and IoU under various noise conditions. The results showed that the model achieved the best performance under normal conditions with a pre-cision of 0,98, recall of 1,00, F1-Score of 0,99, and IoU of 0,84. In the gaussian blur condition, the model showed a precision of 1.00 but a slight decrease in recall to 0,93 and F1-Score 0,96. Darkness and low resolution conditions showed a slight decrease in metrics with an F1-Score of 0,98 each. The motion blur condition was the most challenging with a significant drop in precision to 0,96, recall 0,96, F1-Score 0,96, and IoU 0,80.

.

I. PENDAHULUAN

SAMPAH adalah sisa barang atau bahan yang sudah digunakan oleh manusia. Sampah sering dianggap sebagai sesuatu yang tidak berguna. Secara umum, manusia memandang sampah sebagai barang sisa dari kegiatan manusia yang mengganggu keindahan lingkungan [1]. Pengelompokan sampah di Indonesia adalah langkah penting untuk pengelolaan limbah yang berkelanjutan. Sampah dibagi menjadi beberapa kategori berdasarkan sifat dan

komposisinya. Sampah organik mudah terurai secara alami dan dapat diolah menjadi pupuk kompos. Sampah anorganik sulit terurai dan perlu didaur ulang [2]. Sampah B3 dan e-waste mengandung zat berbahaya dan perlu penanganan khusus [3]. Sampah residu adalah sisa sampah yang tidak termasuk dalam kategori lainnya [4]. Pengelompokan sampah dapat memudahkan proses daur ulang, pengomposan, dan pengurangan sampah yang masuk ke TPA. Selain itu, pengelompokan sampah juga dapat mengedukasi masyarakat tentang pentingnya pengelolaan sampah yang bertanggung jawab.

Sampah di lingkungan sekitar masih tercampur dan tidak terpilah [5]. Hal ini menyebabkan sampah tidak dapat didaur ulang dan menumpuk di Tempat Pembuangan Akhir (TPA). Data dari Sistem Informasi Pengelolaan Sampah Nasional (SIPSN) Kementerian Lingkungan Hidup dan Kehutanan (KLHK) menunjukkan bahwa Indonesia menghasilkan 35,83 juta ton sampah sepanjang tahun 2022. Volume sampah tersebut naik 21,7% dari tahun 2021, sekaligus menjadi yang tertinggi dalam empat tahun terakhir [6]. Peningkatan volume sampah ini disebabkan oleh sampah yang tidak dipilah antara organik dan anorganik [7]. Oleh karena itu dibutuhkan teknologi yang dapat mendeteksi sampah organik dan anorganik. Teknologi ini dapat membantu masyarakat dalam memilah sampah, sehingga sampah dapat didaur ulang dan tidak menumpuk di TPA.

Pengelompokan sampah adalah langkah penting dalam pengelolaan sampah yang efektif. Langkah ini melibatkan penggunaan teknologi atau alat untuk memisahkan sampah berdasarkan jenisnya, seperti organik dan anorganik. Salah satu teknologi yang digunakan untuk pendeteksian sampah adalah mesin pemilah otomatis [8]. Mesin ini menggunakan berbagai metode untuk memisahkan sampah, seperti berdasarkan berat, ukuran, atau bahkan analisis visual. Teknologi ini memungkinkan pemisahan sampah dengan presisi yang tinggi, sehingga material organik dan anorganik dapat dipisahkan secara efisien. Sebelumnya telah dilakukan penelitian oleh Abdurahman Ibnul Rasidi dkk menggunakan metode CNN menghasilkan 96% untuk sampah anorganik dan 62% untuk sampah organik [9].

YOLO (*You Only Look Once*) adalah algoritma deteksi objek secara *real-time* yang menggabungkan pendekatan regresi spasial dan klasifikasi dalam satu arsitektur jaringan saraf tiruan. Algoritma ini membagi gambar menjadi grid dan memprediksi bounding box serta klasifikasi dan memisahkan jenis sampah organik dan anorganik dalam gambar atau video [10]. YOLO pertama kali dibuat oleh Joseph Redmon dan Ali Farhadi pada tahun 2015. YOLO terinspirasi dari *GoogleNet*, yang merupakan model klasifikasi gambar [11].

Sebelumnya sudah ada penelitian yang dilakukan menggunakan YOLOv3 dan ResNet50 tentang Klasifikasi Benda Organik dan Anorganik, dari penelitian tersebut dapat disimpulkan bahwa YOLOv3 mencapai mean Average Precision sebesar 45% dan average loss sebesar 91%. Sementara itu, ResNet50 menggunakan input size 416x416 dengan *learning rate* yang semakin rendah untuk meningkatkan akurasi. Dengan menggabungkan kedua pendekatan ini, ResNet50 dapat meningkatkan akurasi dalam klasifikasi jenis objek yang dideteksi oleh YOLOv3 [12]. Ada pula penelitian mengenai Implementasi YOLOv3 untuk Mengidentifikasi dan Mengklasifikasi Sampah Kantor berbasi NVIDIA Jetson Nano menghasilkan bahwa penggunaan algoritma YOLOv3 dalam sistem klasifikasi sampah perkantoran dapat mencapai akurasi deteksi objek sebesar 94%, akurasi klasifikasi sebesar 97.3%, dan waktu komputasi terbaik sebesar 0.271 detik [13]. Sebelumnya juga ada penelitian yang dilakukan oleh Hendri, dkk menggunakan metode YOLOv3 berbasis Raspberry Pi menghasilkan bahwa sistem dapat mencapai akurasi deteksi sebesar 93.7% untuk sampah anorganik dan 92% untuk sampah organik. Rata-rata waktu komputasi untuk sampah anorganik adalah 12.5433 detik dan untuk sampah organik adalah 15.1685 detik. Pengujian juga menunjukkan tingkat akurasi ketepatan dalam klasifikasi sampah antara 80% hingga 100% [14]. Dari penelitian oleh Heru Purwantoro, dkk [15] menggnakan YOLOv5, menghasilkan hasil nilai yang cukup baik dalam akurasi, dengan nilai akurasi pada merek Teh Pucuk sebesar 87%, merek Aqua 91%, serta merek Le Minerale 92%. Oleh karena itu, dengan menggunakan algoritma YOLOV5, hasil model deteksi yang dihasilkan sudah baik dan dapat digunakan sebagai model deteksi dalam proses penyortiran sampah botol plastik. Dari beberapa penelitian sebelumnya, belum menggunakan *noise* pada citra seperti halnya pada dunia nyata. Maka daripada itu pada penelitian ini mengusulkan untuk menggunakan *noise* pada citra.

Menurut penelitian yang dilakukan oleh Murat Tasyürek dan Celal Öztürk [16] mengusulkan bahwa pendekatan berbasis CNN untuk deteksi nomor rumah dari gambar real-time, menguji model Faster R-CNN, MobileNet, YOLOv4, YOLOv5, dan YOLOv7 pada gambar dari Provinsi Kayseri. Model-model ini menunjukkan peningkatan performa setelah di-fine-tune, dengan YOLOv5 mencapai skor f1 tertinggi 0.972, dan YOLOv7 memiliki waktu deteksi tercepat 0.009 detik. Dari hasil ini, kita dapat mengasumsikan bahwa YOLOv8 akan unggul dibandingkan YOLOv7 karena beberapa alasan teknis. YOLOv8 kemungkinan memiliki arsitektur yang lebih efisien dan canggih dibandingkan YOLOv7, yang akan meningkatkan akurasi dan kecepatan deteksi lebih lanjut. Selain itu, YOLOv8 mungkin menyertakan mekanisme pengolahan data yang lebih baik, memungkinkan model untuk menangani variasi kedalaman

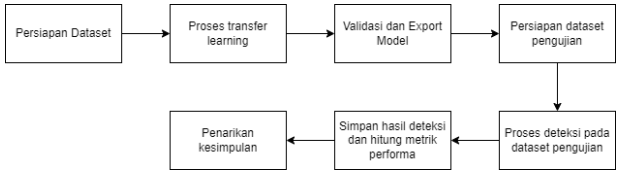
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

dan kondisi pencahayaan dalam gambar dengan lebih efektif. Berdasarkan kesimpulan penelitian, penggunaan YOLOv8 dapat diharapkan memberikan hasil yang lebih unggul dalam hal akurasi dan kecepatan, menjadikannya lebih andal dan cepat dalam mendeteksi nomor rumah dari gambar real-time yang bervariasi.

Namun belum ada penelitian menggunakan YOLOv8 untuk deteksi sampah organik dan anorganik dengan berbagai macam skenario seperti filter normal, *gaussian blur*, *darkness*, *low resolution*, dan *motion blur*. Maka daripada itu penelitian ini bertujuan untuk meningkatkan akurasi dalam identifikasi sampah organik dan anorganik menggunakan YOLOv8 dan penambahan noise pada citra. YOLOv8, yang membangun kekuatan dari pendahulunya, kemungkinan akan melanjutkan tren performa tinggi dalam tugas deteksi objek secara real-time. Model YOLO secara konsisten mengungguli model Faster R-CNN dan SSD dalam hal kecepatan, dengan tetap mempertahankan tingkat akurasi dan presisi yang tinggi [17]. Untuk aplikasi yang memerlukan deteksi real-time dan perangkat tenaga rendah, model YOLO, terutama versi lightweight seperti YOLOv8, sangat cocok digunakan untuk mendeteksi sampah di Indonesia.

II. METODE PENELITIAN

Ada beberapa tahapan penting yang harus dilewati selama proses pengembangan model YOLOv8 untuk deteksi sampah organik dan anorganik. Tahapan-tahap ini termasuk persiapan dataset, transfer learning, evaluasi dan export model, persiapan dataset untuk pengujian, proses deteksi, menyimpan hasil deteksi, dan menghitung metrik performa. Pada tahapan tersebut dapat dilihat pada Gambar 1.



Gambar. 1. Alur Penelitian

A. *Dataset*

Langkah awal melibatkan pengumpulan dan penyusunan dataset gambar yang mencakup berbagai jenis sampah organik dan anorganik [18]. Setiap gambar perlu dilabeli sesuai dengan jenis sampahnya. Dataset yang digunakan ini bersumber dari *Roboflow* yang di mana data ini terdiri dari 2 kelas yaitu organik dan anorganik, dari masing-masing kelas tersebut terbagi menjadi 10 kategori. Dataset ini kemudian akan digunakan untuk melatih model YOLOv8.

Dalam dataset yang digunakan ini, tidak dilakukan modifikasi terhadapnya, dikarenakan dataset ini sudah di proses oleh Roboflow dengan sudah dilakukanya pemecahan dan distribusi set kedalam bagian train, test dan valid.

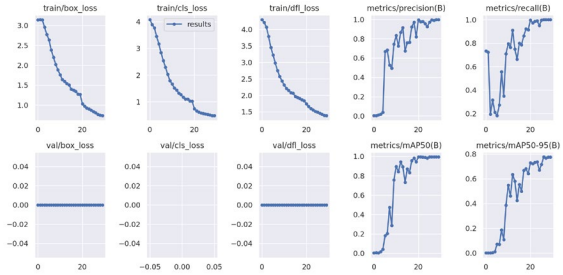
Selain itu, model YOLOv8 yang akan digunakan dalam proses transfer learning juga perlu dipersiapkan. Model ini dapat berupa model pra-terlatih yang akan disesuaikan dengan dataset yang telah disiapkan.

B. *Proses Transfer Learning*

Proses *transfer learning* dilakukan dengan memanfaatkan dataset yang telah disiapkan untuk melatih model YOLOv8 [19]. Pada tahap ini parameter model disesuaikan dengan karakteristik dataset, sehingga model dapat mendeteksi sampah organik dan anorganik dengan akurasi tinggi. *Transfer learning* memungkinkan penggunaan yang diperoleh dari model pra-terlatih dan menyesuaikannya dengan dataset baru yang spesifik. Pada tahap ini dilakukan selama 30 *epoch*, dan ukuran input 640x640 untuk memaksimalkan performa.

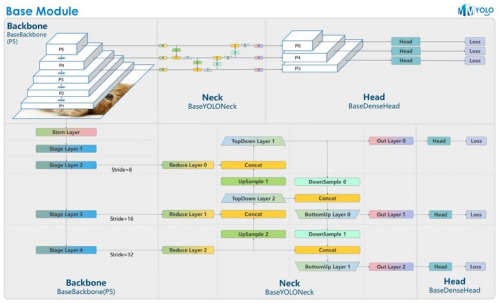
Angka parameter transfer learning ditentukan berdasarkan beberapa percobaan pelatihan dalam penelitian ini. Ditemukan bahwa angka *epoch* yang optimal adalah 25-30, karena performa model memuncak di *epoch* 25 dan tidak mengalami peningkatan signifikan setelahnya. Grafik pelatihan menunjukkan bahwa model mencapai performa tertinggi pada *epoch* 25.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506



Gambar. 2. Grafik Proses Pelatihan (Ultralytics)

Arsitektur YOLOv8 terdiri dari dua bagian utama, *Backbone* dan *Head/Prediction*, yang digunakan untuk proses *transfer learning*.



Gambar. 3. Arsitektur YOLOv8 (17)

Backbone, juga disebut sebagai ekstraktor fitur, bertugas mengekstrak fitur penting dari input. Pola sederhana di lapisan awal, seperti tepi dan tekstur, ditangkap di bagian ini. *Neck* berfungsi sebagai penghubung antara *head* dan *backbone*, melakukan operasi fusi fitur, dan mengintegrasikan informasi kontekstual. Bagian akhir jaringan adalah *head*, yang bertanggung jawab untuk menghasilkan output seperti kotak pembatas dan skor kepercayaan untuk deteksi objek. *Head* menghasilkan kotak pembatas yang terkait dengan objek yang mungkin ada dalam gambar, menetapkan skor kepercayaan untuk setiap kotak pembatas untuk menunjukkan seberapa mungkin suatu objek ada, dan mengurutkan objek dalam kotak pembatas menurut kategori mereka [20]. Arsitektur YOLOv8 dapat dilihat pada Gambar 3.

C. *Validasi dan Export Model*

Setelah proses tranfer learning selesai, model dievaluasi untuk mengukur kinerjanya. Evaluasi ini dilakukan menggunakan dataset validasi untuk menilai berbagai metrik performa seperti akurasi, presisi, recall, dan mAP (mean Average Precision). Model yang telah dievaluasi kemudian diekspor dalam bentuk model PyTorch atau .pT diambil best pt dengan akurasi yang terbaik. Setelah proses transfer learning selesai dan model diekspor (best.pt), evaluasi dilakukan pada bagian test dari dataset pelatihan. Hasil evaluasi ini memberikan metrik yang mewakili performa model terhadap bagian test dari dataset pelatihan.

D. *Pengujian*

Dataset pengujian disiapkan untuk menilai performa model yang sudah dilatih dan diekspor. Dataset ini harus berbeda dari dataset yang digunakan untuk pelatihan dan validasi agar memastikan model dapat mendeteksi sampah pada data yang belum pernah dilihat sebelumnya.

Dataset pengujian akan dibuat menjadi beberapa variasi dengan filter-filter gambar yang berbeda seperti normal, *gaussian blur*, *motion blur*, *low resolution*, dan *darkness*. Variasi-variasi ini memiliki fungsi agar metrik metrik dari hasil pengujian dapat dilakukan perbandingan dengan kasus pengujian filter yang lainnya untuk mengetahui karakteristik dari model dalam kasus pengujian tersebut.

Untuk validitas pengujian, dataset diuji dengan variasi filter gambar: normal, *gaussian blur*, *motion blur*, *low resolution*, dan *darkness*. *Gaussian blur* mensimulasikan buram akibat gerakan atau fokus tidak tepat. *Motion blur*

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

mensimulasikan buram akibat gerakan cepat atau pengambilan gambar tidak stabil. *Low resolution* menurunkan resolusi gambar untuk mensimulasikan kualitas rendah. *Darkness* mengurangi kecerahan gambar untuk mensimulasikan kondisi pencahayaan rendah. Filter ini relevan dengan kondisi dunia nyata karena model deteksi objek sering beroperasi dalam variasi kualitas gambar. Menguji model pada variasi ini mengevaluasi kemampuan model dalam kondisi berbeda, memberikan pemahaman lebih baik tentang performa model dalam kondisi sebenarnya. Pada Gambar 4 sampai 8 merupakan variasi dari dataset yang diberikan filter

Gambar. 4 *Normal*

Gambar. 5 *Darkness*

Gambar. 6 *Gaussian Blur*

Gambar. 7 *Low Resolution*

Gambar. 8 *Motion Blur*

E. Proses Deteksi Pada Dataset Pengujian

Model YOLOv8 yang telah diekspor digunakan untuk mendeteksi sampah pada gambar-gambar di dataset pengujian. Proses deteksi ini menghasilkan output berupa bounding box dan label untuk setiap objek sampah yang terdeteksi. Pada proses ini juga diulangi pada variasi-variasi kasus pengujian yang ada.

F. Simpan Hasil Deteksi dan Hitung Metrik Performa

Hasil deteksi dari dataset pengujian disimpan untuk analisis lebih lanjut. Selain itu, metrik performa seperti *precision*, *recall*, *mAP*, dan *IoU* dihitung untuk menilai efektivitas model dalam mendeteksi sampah organik dan anorganik.

Dalam penelitian ini dilakukan analisa terhadap metrik metrik performa model antara lain mAP dan IoU. mAP bisa digunakan untuk semua case object detection tidak spesifik hanya YOLO saja, bisa digunakan untuk model object detection seperti Faster R-CNN, mobilenet, dan lain-lain. F1-Score juga bisa digunakan untuk YOLO [21]. Penggunaan mAP dalam evaluasi model YOLO lebih disukai karena memberikan penilaian yang lebih komprehensif dan akurat tentang kinerja model dalam mendeteksi objek di berbagai kondisi dan kelas. Sementara F1-Score tetap penting, terutama dalam konteks di mana keseimbangan antara precision dan recall sangat penting, mAP menawarkan keunggulan dalam evaluasi model deteksi objek secara keseluruhan.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Setelah metrik *precission* berhasil dihitung, maka selanjutnya metrik *mAP* dapat dihitung dengan rumus.

$$AP = \sum(recall_{n+1} - recall_n).precision_{interp}.(recall_{n+1}) \quad (3)$$

Metrik *IoU* dapat dihitung dengan rumus dibawah ini.

$$IoU = \frac{area\ (BB_{prediksi} \cap BB_{groundTruth})}{area(BB_{prediksi} \cup BB_{groundTruth})} \quad (4)$$

Keterangan:
BB = Bounding Box

Dengan semakin tingginya angka metrik *IoU* dari sebuah model, maka semakin tepat prediksi lokasi objek dibandingkan dengan lokasi pada label yang sebenarnya.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

G. *Spesifikasi Hardware*

Dalam pengimplementasian penelitian ini, diperlukan perangkat keras komputer yang menggunakan arsitektur x64 dengan spesifikasi minimal tertentu. Spesifikasi ini menjadi kebutuhan kritis untuk menjalankan model dan memberikan landasan evaluasi kinerja kecepatan model dalam konteks penelitian ini.

TABEL I
SPESIFIKASI *HARDWARE*

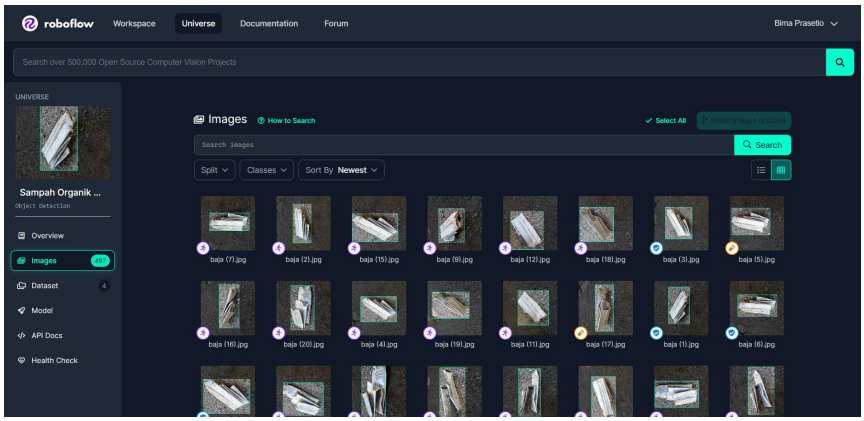
No	Hardware	Spesifikasi
1.	*Processor*	Minimal Kecepatan 1 GHz / *Intel Core i5-4700*
2.	*RAM*	8 GB
3.	*GPU*	Mendukung *CUDA* dengan RAM 4GB / *GTX 1050 TI*
4.	Penyimpanan	256 GB
5.	Sistem Operasi	*Windows 8* / *Ubuntu 16.0* / *MacOS 10.12.6* / *WSL 2*

III. HASIL DAN PEMBAHASAN PENELITIAN

Dalam bab ini akan menjelaskan terkait apa saja tentang proses pengembangan dan hasilnya.

A. *Spliting Dataset*

Dalam penelitian ini, dataset yang digunakan untuk melatih model YOLOv8 terdiri dari gambar-gambar berbagai jenis sampah organik dan nonorganik. Dataset ini diperoleh dari *Roboflow* yang dibuat oleh SISCER-Project dan dibagi menjadi dua kelas utama: organik dan anorganik, dengan jumlah total citra 1178 dan dibagikan menjadi *train* 87% (1024 citra), *valid* 8% (98 citra), *test* 5% (56 citra). Seluruh dataset dapat dilihat pada Gambar 9.



Gambar. 9. Dataset Sampah Organik dan Anorganik

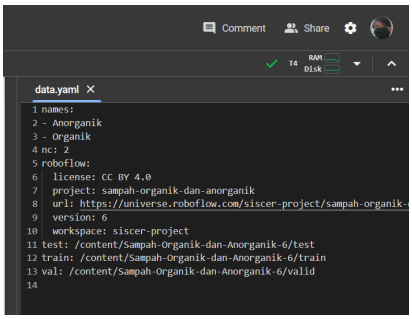
Dari dataset yang sudah didownload dapat digambarkan sebagai berikut. Dataset dibagi menjadi 3 bagian, yaitu train, test, dan valid, dan dari masing-masing bagian dipisahkan menjadi 2 direktori yaitu images dan labels.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

```
|-- dataset
|   |-- test
|   |   |-- images
|   |   |   |-- gambar.jpg
|   |   |-- labels
|   |   |   |-- label_YOLO.txt
|   |-- train
|   |   |-- images
|   |   |   |-- gambar.jpg
|   |   |-- labels
|   |   |   |-- label_YOLO.txt
|   |-- valid
|   |   |-- images
|   |   |   |-- gambar.jpg
|   |   |-- labels
|   |   |   |-- label_YOLO.txt
|   |-- data.yaml
```

Gambar. 10. Struktur Folder Dataset

Keseluruhan dataset ini terkandung didalam direktori ‘dataset’ yang di mana mengandung keseluruhan dari bagian-bagian dataset dan juga data.yaml yang mengandung informasi terkait dataset seperti direktori utama direktori pecahan, dan jumlah nama kelas yang dapat digunakan oleh proses YOLO dalam pelatihan.



Gambar. 11. Data.yaml

B. *Transfer Learning Model*

Proses transfer learning dalam penelitian ini menggunakan model YOLOv8 pra-terlatih yang dilatih kembali dengan dataset baru berisi gambar sampah organik dan anorganik dari Roboflow. Pelatihan berlangsung selama 30 *epoch* dengan ukuran gambar 640x640 piksel. Dalam proses training menggunakan YOLOv8m model ini memiliki jumlah layers dengan total sebanyak 295 layers, mempunyai 25,857,478 parameter yang diatur dalam model YOLOv8m, dapat dilihat pada Gambar 12 dibawah ini.

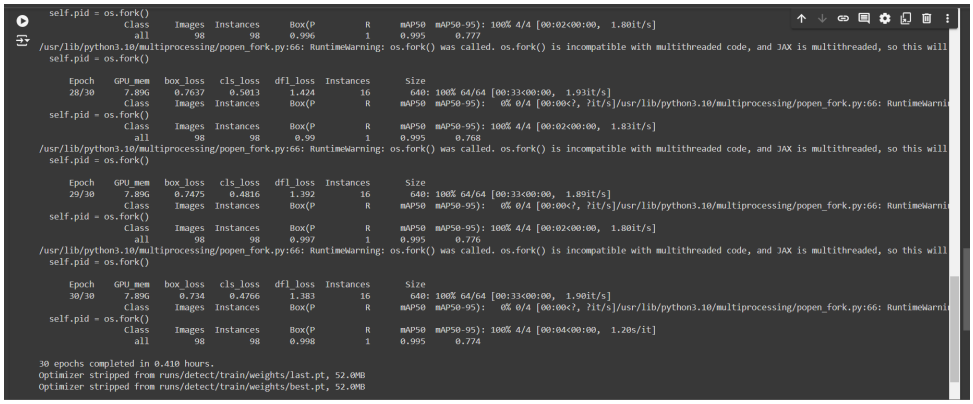
	from	n	params	module	arguments
0	-1	1	1392	ultralytics.nn.modules.Conv	[3, 48, 3, 2]
1	-1	1	41664	ultralytics.nn.modules.Conv	[48, 96, 3, 2]
2	-1	2	111360	ultralytics.nn.modules.C2f	[96, 96, 2, True]
3	-1	1	166272	ultralytics.nn.modules.Conv	[96, 192, 3, 2]
4	-1	4	813312	ultralytics.nn.modules.C2f	[192, 192, 4, True]
5	-1	1	664320	ultralytics.nn.modules.Conv	[192, 384, 3, 2]
6	-1	4	3248640	ultralytics.nn.modules.C2f	[384, 384, 4, True]
7	-1	1	1991808	ultralytics.nn.modules.Conv	[384, 576, 3, 2]
8	-1	2	3985920	ultralytics.nn.modules.C2f	[576, 576, 2, True]
9	-1	1	831168	ultralytics.nn.modules.SPPF	[576, 576, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.Concat	[1]
12	-1	2	1993728	ultralytics.nn.modules.C2f	[960, 384, 2]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.Concat	[1]
15	-1	2	517632	ultralytics.nn.modules.C2f	[576, 192, 2]
16	-1	1	332160	ultralytics.nn.modules.Conv	[192, 192, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.Concat	[1]
18	-1	2	1846272	ultralytics.nn.modules.C2f	[576, 384, 2]
19	-1	1	1327872	ultralytics.nn.modules.Conv	[384, 384, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.Concat	[1]
21	-1	2	4207104	ultralytics.nn.modules.C2f	[960, 576, 2]
22	[15, 18, 21]	1	3776854	ultralytics.nn.modules.Detect	[2, [192, 384, 576]]

Model summary: 295 layers, 25857478 parameters, 25857462 gradients, 79.1 GFLOPs

Gambar. 12. Parameter YOLOv8

Setelah konfigurasi parameter training lalu dilakukan pelatihan transfer learning selama 30 *epoch*, sesuai dengan konfigurasi yang diatur. Proses training dilakukan menggunakan Google Colaboratory dengan perangkat keras *Nvidia Tesla T4*. Proses training dilakukan selama 24 menit.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506



Gambar. 13. Training Model YOLOv8m

Setelah proses training selesai, maka selanjutnya dilakukan proses validasi terhadap model yang selesai dilatih.

C. *Validasi Model*

Setelah proses transfer learning selesai, model dapat dievaluasi untuk mengukur bagaimana kinerja dari model yang sudah dilatih. Evaluasi ini dilakukan menggunakan dataset validasi untuk menilai berbagai metrik performa seperti akurasi, *precision*, *recall*, dan *mean Average Precision* (*mAP*). Model yang sudah di validasi kemudian dieskpor dalam bentuk model PyTorch (.pt) dan diambil model dengan akurasi terbaik. Dari hasil validasi dapat dilihat dengan angka-angka berikut.

TABEL II
ANGKA-ANGKA VALIDASI

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
All	98	98	0.996	1	0.995	0.778
Anorganik	98	43	0.999	1	0.995	0.784
Organik	98	55	0.993	1	0.995	0.771

Dari validasi dilakukan dengan jumlah 98 gambar, menghasilkan *precision* (P) sebesar 0.996, *recall* (R) sebesar 1.0, *mAP@50* sebesar 0.995, dan *mAP@50-95* sebesar 0.778 untuk keseluruhan kelas. Kelas anorganik dan organik masing-masing menunjukkan *precision* sebanyak 0.999 dan 0.993, *recall* sebanyak 1.0, *mAP@50* sebesar 0.995, dan *mAP@50-95* sebesar 0.784 dan 0.771. Kecepatan proses inferensi mencapai 8.9ms per gambar.

D. *Pengujian Deteksi*

Setelah dilakukan training dan validasi model yang bagus, menghasilkan angka validasi yang memuaskan maka dapat dilanjutkan proses pengujian yang dilakukan dengan dataset yang nyata dengan menggunakan script python dengan menggunakan model yang sudah tersimpan. Proses pengujian dapat dipecahkan menjadi 4 bagian, sebagai berikut:

1. Inisialisasi model YOLOv8
2. Memuat gambar-gambar yang akan diuji dari direktori dataset pengujian
3. Melakukan deteksi pada objek setiap gambar
4. Menyimpan hasil deteksi (termasuk bounding box, nama kelas, dan confidence score) ke dalam sebuah file JSON.
5. Simpan hasil pengujian ke bentuk excel
6. Proses Pengulangan deteksi untuk tiap variasi

Proses pengujian dapat dirincikan sebagai berikut:

1. Inisialisasi Model YOLOv8: Pada proses inisialisasi model YOLOv8 *script* memuat model YOLOv8 yang telah dilatih sebelumnya dari file “best3.pt”. Model ini digunakan untuk mendeteksi objek dalam gambar.
2. Memuat Gambar dari Direktori: Setelah model sudah dimuat dalam memori program, maka selanjutnya perlu dilakukan memuat gambar gambar yang akan diuji, gambar-gambar ini berasal dari dataset pengujian yang sudah dipersiapkan sebelumnya, dataset pengujian ini memiliki struktur direktori yang sama seperti direktori train.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

```
|   |-- test
|   |   |-- images
|   |   |   |-- gambar.jpg
|   |   |-- labels
|   |   |   |-- label_YOLO.txt
```

Gambar. 14. Struktur Folder Test

Dataset pengujian memiliki format yang mirip dengan dataset pelatihan, di mana memiliki direktori images dan labels. Di mana memerlukan labels dikarenakan memiliki peran sebagai pembanding hasil deteksi dengan label asli. Dari gambar-gambar yang sudah dimuat dari dataset pengujian dimuat juga label-label dari gambar tersebut, untuk dilakukan perbandingan, sebelum itu perlu dilakukan deteksi terhadap dataset pengujian.

3. Proses Deteksi:
    a. Preprocess: Dalam proses deteksi, script memuat daftar gambar dari direktori dan membaca setiap gambar menggunakan OpenCV ("cv2.imread").
    b. Inference: Model YOLO kemudian memprediksi objek dalam gambar, menghasilkan koordinat bounding box, confidence score, dan id kelas objek.
    c. Postprocess: Memproses hasil deteksi untuk mendapatkan bounding boxes dalam format yang sesuai, confidence scores, dan nama kelas objek.
    d. Menyimpan hasil terbaik: Koordinat bounding box disesuaikan dengan ukuran gambar asli. Deteksi dengan tingkat kepercayaan tertinggi diekstraksi, dan informasi ini (termasuk nama kelas objek, bounding box, dan tingkat kepercayaan) disimpan dalam dictionary hasil. Waktu untuk setiap tahap (preprocessing, inference, postprocessing) juga dicatat. Pada Gambar 16 adalah contoh hasil dari deteksi, yang di mana bounding box berwarna biru adalah bounding box dari label asli, bounding box berwarna merah adalah bounding box dari hasil deteksinya. Pada tahap ini dilakukan juga untuk variasi-variasi yang ada. Dari dataset pengujian data tersebut merupakan memiliki kelas anorganik, untuk hasil deteksi menghasilkan kelas deteksi anorganik.

Gambar. 15. Contoh Gambar yang sudah dideteksi

4. Menyimpan Hasil ke File JSON: Proses penyimpanan hasil dalam script YOLOv8 dilakukan dengan mengumpulkan informasi deteksi dari setiap gambar yang diproses. Informasi ini mencakup nama file gambar, kelas objek yang terdeteksi, koordinat bounding box, confidence score, dan waktu pemrosesan. Semua data deteksi ini disimpan dalam sebuah dictionary. Setelah seluruh gambar diproses, dictionary ini kemudian disimpan sebagai file JSON di direktori hasil yang telah ditentukan, memastikan bahwa semua hasil deteksi tersimpan dalam format yang terstruktur dan mudah diakses untuk analisis lebih lanjut.
5. Simpan hasil pengujian ke bentuk Excel: Proses ini menyimpan hasil dari pengujian beserta metrik metrik hasil deteksi kedalam format tabel dalam suatu dokumen excel. Data yang tersimpan pada dokumen excel ini adalah: Nama Gambar, Kelas asli sesuai label, Kelas yang di deteksi, *confi-dence score*, evaluasi *confusion matrix* (TP, FP, TN, FN), *IoU*, dan waktu pemrosesan. Dokumen excel ini mempermudah dalam proses analisis yang dapat langsung diakses.
6. Proses pengulangan deteksi untuk tiap variasi: Keseluruhan proses deteksi ini akan dilakukan untuk semua variasi dari kasus pengujian, yang bertujuan untuk mengetahui keunggulan dan keku-rangan dari model dalam proses deteksi untuk masing masing kasus pengujian, dan juga untuk mengetahui karakteristik model.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

E. Pengumpulan Hasil

Script ini dirancang untuk mengevaluasi hasil deteksi objek dari YOLOv8 dengan menghitung metrik seperti *precision*, *recall*, *mean Average Precision* (*mAP*), dan *Intersection over Union* (*IoU*). Pada tahap awal, script membaca label YOLO yang berisi kelas objek dan bounding box menggunakan fungsi "read_yolo_label", yang mengubah label-label ini menjadi format dictionary dan menyimpannya dalam daftar. Fungsi "convert_bbox_yolo_to_coners" kemudian digunakan untuk mengonversi bounding box dari format YOLO yang berpusat di tengah menjadi format sudut (xmin, ymin, xmax, ymax).

Selanjutnya, script menggunakan fungsi "compute_iou" untuk menghitung *Intersection over Union* (*IoU*) antara dua bounding box, yang merupakan metrik utama untuk mengevaluasi seberapa baik prediksi sesuai dengan ground truth. Fungsi "parse_json_results" membaca file JSON yang berisi hasil deteksi dari YOLOv8 dan mengembalikan daftar deteksi serta direktori gambar. Ukuran gambar diperoleh melalui fungsi "get_image_size", yang membaca gambar dan mengembalikan lebar serta tingginya.

Bagian utama dari script ini adalah fungsi "calculate_metrics", yang menghitung berbagai metrik evaluasi seperti *precision*, *recall*, *mAP*, *IoU* rata-rata, serta confusion matrix (TP, FP, FN, TN). Script ini membaca hasil deteksi dan label *ground truth*, mengonversi bounding box, dan menghitung IoU untuk setiap pasangan deteksi dan label. Berdasarkan IoU, script menentukan true positive (TP), false positive (FP), dan false negative (FN). *Precision*, *recall*, dan *mAP* dihitung bersama dengan *IoU* rata-rata dan *confusion matrix*.

Selanjutnya script menunjukkan contoh penggunaan dari fungsi-fungsi di atas untuk mengevaluasi hasil deteksi dari file JSON dan direktori label. Hasil evaluasi, termasuk mAP, precision, recall, dan *confusion matrix*, dicetak ke konsol, memberikan gambaran lengkap tentang kinerja model deteksi objek.

Pada akhirnya script ini membuat dokumen excel yang berisikan nama gambar, kelas asli sesuai label, kelas yang di deteksi, *confidence score*, evaluasi *confusion matrix* (TP, FP, TN, FN), *IoU*, dan waktu pem-rosesan, yang dapat di lakukan analisa secara langsung oleh manusia.

Hasil hasil berupa JSON dan Excel dari semua variasi kasus pengujian akan dikumpulkan, dan metrik metrik performa dari hasil deteksi akan dilakukan perbandingan untuk menentukan keunggulan atau kekurangan dalam kasus pengujian tertentu dan untuk mengetahui karakteristik model.

F. Analisis Hasil

Analisis kinerja model deteksi dalam berbagai kasus. Tabel yang disediakan menunjukkan kinerja model deteksi dalam berbagai kondisi pengujian, termasuk pengujian normal, gaussian blur, darkness, low resolution, dan motion blur. Berikut adalah analisis menyeluruh untuk setiap kondisi.

TABEL III
KINERJA MODEL DETEKSI DALAM BERBAGAI VARIASI

Kasus	*Precision*	*Recall*	*mAP*	*F1 Score*	*IoU*	Rata-Rata Waktu Pemrosesan (ms)
Normal	0,98	1,00	0,98	0,99	0,84	61,26
Gaussian Blur	1,00	0,93	1,00	0,96	0,79	60,23
Darkness	0,98	0,98	0,98	0,98	0,83	75,03
Low Resolution	0,98	0,98	0,98	0,98	0,83	61,81
Motion Blur	0,96	0,96	0,96	0,96	0,80	60,45

1. Normal

TABEL IV
*CONFUSION MATRIX*NORMAL

Kasus	TP	TN	FP	FN	Total Image
Normal	55	0	1	0	56

$$Precision = \frac{TP}{TP+FP} = \frac{55}{55+1} = 0{,}98$$

$$Recall = \frac{TP}{TP+FN} = \frac{55}{55+0} = 1$$

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 494-506

Dalam kondisi normal, model deteksi beroperasi dengan sangat baik, dengan nilai *precision* 0,98, yang menunjukkan bahwa 98% dari semua deteksi yang dilakukan oleh model adalah benar. *Recall* mencapai nilai sempurna 1,00, yang menunjukkan bahwa model berhasil mendeteksi semua objek yang ada tanpa melewatkan satupun. Kinerja yang baik secara keseluruhan ditunjukkan oleh nilai *mean Average Precision* (*mAP*) 0,98. Dengan *F1-Score* 0,99, yang merupakan keseimbangan antara *precision* dan *recall*, hasilnya menunjukkan performa yang hampir sempurna. Akurasi lokal yang baik dan tumpang tindih yang signifikan antara kotak prediksi dan *ground truth* menunjukkan *Intersection over Union* (*IoU*) sebesar 0,84. Waktu pemrosesan rata-rata (61,26 ms) adalah standar untuk kondisi lainnya.

2. *Gaussian Blur*

TABEL V
CONFUSION MATRIX GAUSSIAN BLUR

Kasus	TP	TN	FP	FN	Total Image
Gaussian Blur	52	0	0	4	56

$$Precision = \frac{TP}{TP+FP} = \frac{52}{52+0} = 1$$

$$Recall = \frac{TP}{TP+FN} = \frac{52}{52+4} = 0{,}93$$

Dalam kondisi Gaussian Blur, model tetap menunjukkan precision sempurna sebesar 1,00, yang berarti semua deteksi adalah benar tanpa *false positive*. Namun, *recall* turun menjadi 0,93, yang menunjukkan bahwa model melewatkan sekitar 7% objek yang sebenarnya ada. Nilai *mAP* tetap sempurna di 1,00, yang menunjukkan bahwa ketika model mendeteksi objek, deteksi tersebut dilakukan dengan sangat akurat dan percaya diri. Ada perbedaan antara ketepatan dan *recall*, menurut *F1 Score* sebesar 0,96. Waktu pemrosesan rata-rata sedikit lebih cepat, yaitu 60,23 ms, dan akurasi lokal sedikit terganggu oleh *Gaussian Blur*, karena *IoU* turun menjadi 0,79.

3. *Darkness*

TABEL VI
CONFUSION MATRIX DARKNESS

Kasus	TP	TN	FP	FN	Total Image
Darkness	54	0	1	1	56

$$Precision = \frac{TP}{TP+FP} = \frac{54}{54+1} = 0{,}98$$

$$Recall = \frac{TP}{TP+FN} = \frac{54}{54+1} = 0{,}98$$

Pada kondisi gelap, ketepatan model 0,98, yang menunjukkan bahwa 98% deteksi adalah benar, dan *recall* juga 0,98, yang menunjukkan bahwa model berhasil mendeteksi 98% objek, hanya melewatkan beberapa. Nilai *mAP* sebesar 0,98 menunjukkan kinerja yang baik secara keseluruhan bahkan dalam kondisi kurang cahaya. *F1 Score* tetap tinggi sebesar 0,98, menunjukkan keseimbangan yang baik antara ketepatan dan *recall*. Dengan *IoU* sebesar 0,83, akurasi lokal menurun, menunjukkan bahwa kotak prediksi kurang tepat dalam kondisi gelap. Sebagai bukti bahwa model memerlukan waktu yang lebih lama untuk memproses gambar dalam kondisi kurang cahaya, waktu rata-rata pemrosesan meningkat secara signifikan menjadi 75,03 ms.

4. *Low Resolution*

TABEL VII
CONFUSION MATRIX LOW RESOLUTION

Kasus	TP	TN	FP	FN	Total Image
Low Resolution	54	0	1	1	56

$$Precision = \frac{TP}{TP+FP} = \frac{54}{54+1} = 0{,}98$$

$$Recall = \frac{TP}{TP+FN} = \frac{54}{54+1} = 0{,}98$$

Model tetap memiliki tingkat akurasi tinggi sebesar 0,98 bahkan dalam kondisi resolusi rendah. *Recall* sebesar 0,98 juga menunjukkan kemampuan deteksi yang kuat meskipun dalam resolusi yang lebih rendah. Nilai *mAP* sebesar 0,98 menunjukkan bahwa kinerja tetap stabil dan baik secara keseluruhan. Sebagai bukti keseimbangan yang baik antara ketepatan dan ingat, *F1 Score* tetap pada 0,98. Akurasi lokal yang sedikit menunjukkan penurunan, sebanding dengan kondisi gelap, dengan *IoU* sebesar 0,83. Waktu pemrosesan rata-rata 61,18 ms, hampir sama dengan kondisi normal, menunjukkan bahwa model ini cukup baik untuk menangani gambar beresolusi rendah.

5. *Motion Blur*

TABEL VIII
CONFUSION MATRIX MOTION BLUR

Kasus	TP	TN	FP	FN	Total Image
Motion Blur	52	0	2	2	56

$$Precision = \frac{TP}{TP+FP} = \frac{52}{52+2} = 0{,}96$$

$$Recall = \frac{TP}{TP+FN} = \frac{52}{52+2} = 0{,}96$$

Efek *motion blur* menimbulkan tantangan besar bagi model, dengan precision menurun menjadi 0,96, yang menunjukkan peningkatan *false positive*. *Recall* juga turun menjadi 0,96, yang berarti ada lebih banyak objek yang tidak terdeteksi. Nilai *mAP* sebesar 0,96 menunjukkan penurunan kinerja dalam rata-rata precision. *F1 Score* sebesar 0,96 menunjukkan penurunan kinerja keseluruhan. *IoU* sebesar 0,80 merupakan yang terendah di antara semua kondisi, menandakan tantangan besar dalam akurasi lokal. Waktu pemrosesan rata-rata adalah 60,45 ms, yang sedikit lebih cepat dari kondisi normal tetapi tidak signifikan.

Motion Blur mempengaruhi performa model dalam mendeteksi sampah dalam gambar, hal ini dikarenakan hilangnya atau turunnya definisi dari tepi (*edge*) dari objek tersebut, dikarenakan YOLO merupakan model CNN yang memiliki proses konvolusi (*convolution*), maka adanya perubahan terhadap definisi atau bentuk objek akan mempengaruhi bagaimana model mendeteksi dan mengenali objek tersebut. *Motion Blur* merupakan fenomena di mana suatu kamera mengambil gambar dari objek yang bergerak lebih cepat dari kecepatan *shutter* (*shutter speed*) dari kamera tersebut (atau sebaliknya kamera bergerak lebih cepat dari *shutter speed* dalam satu *frame* tersebut), yang menghasilkan *blur* (buram) yang memiliki arah yang sama dengan arah pergerakan objek atau kamera. *Motion Blur* di dunia nyata dapat diatasi dengan memastikan kamera dan objek diam di satu tempat, atau dalam kasus pemilahan sampah yaitu memastikan objek bergerak dalam kecepatan yang stabil (diatas konveyor) dan memastikan kamera memiliki *shutter speed* yang memadai [22].

Dalam penelitian yang dilakukan oleh Abdurrahman Ibnul Rasidi, metode Convolutional Neural Network (CNN) digunakan untuk mengklasifikasikan sampah organik dan anorganik, dengan hasil ketepatan 96% untuk sampah anorganik dan 62% untuk sampah organik [9]. Sebaliknya, penelitian ini menggunakan model YOLOv8 untuk deteksi sampah organik dan anorganik, dengan hasil precision 0,98 dan recall 1,00 dalam kondisi normal, yang menunjukkan kinerja yang hampir sempurna dengan F1-Score 0,99 dan mean Average Precision (mAP) 0,98. Meskipun model ini juga diuji dalam berbagai kondisi seperti Gaussian Blur, pencahayaan rendah, dan resolusi rendah, tetap menunjukkan hasil yang konsisten tinggi, meskipun dengan sedikit penurunan performa, terutama dalam kondisi motion blur dengan precision dan recall masing-masing 0,96 dan IoU 0,80. Waktu pemrosesan rata-rata untuk model YOLOv8 adalah sekitar 60 ms, sedikit lebih lambat dalam kondisi kurang cahaya. Secara keseluruhan, meskipun kedua penelitian menunjukkan kinerja yang baik, model YOLOv8 menunjukkan hasil yang lebih superior dan konsisten di berbagai kondisi dibandingkan dengan metode CNN yang digunakan oleh Rasidi, yang memiliki performa lebih bervariasi terutama pada deteksi sampah organik.

Berdasarkan hasil analisis dan evaluasi, model ini dapat diimplementasikan dalam sistem pemilahan sampah otomatis. Dengan menggunakan model ini, perangkat deteksi seperti komputer atau Jetson Nano, kamera, dan konveyor dapat digunakan.

IV. KESIMPULAN

Hasil evaluasi menunjukkan performa model YOLOv8 dalam mendeteksi sampah organik dan anorganik melalui evaluasi metrik precision, recall, F1-Score, mean Average Precision (mAP), dan Intersection over Union (IoU) dalam berbagai kondisi pengujian. Hasil pengujian menunjukkan bahwa dalam kondisi normal, model YOLOv8 mencapai precision sebesar 0,98, recall 1,00, F1-Score 0,99, mAP 0,98, dan IoU 0,84. Pada kondisi Gaussian Blur, nilai precision mencapai 1,00, recall 0,93, F1-Score 0,96, mAP 1,00, dan IoU 0,79. Dalam kondisi Darkness, precision dan recall masing-masing sebesar 0,98, F1-Score 0,98, mAP 0,98, dan IoU 0,83. Pada Low Resolution, precision dan recall tetap 0,98, dengan F1-Score 0,98, mAP 0,98, dan IoU 0,83. Sementara itu, pada kondisi Motion Blur, terdapat penurunan performa yang terlihat dari nilai precision 0,96, recall 0,96, F1-Score 0,96, mAP 0,96, dan IoU 0,80. Kesimpulannya, model YOLOv8 menunjukkan kinerja yang sangat baik dalam kondisi normal dan tetap konsisten tinggi dalam kondisi Gaussian Blur, Darkness, dan Low Resolution. Namun, terdapat penurunan performa dalam kondisi Motion Blur yang memengaruhi precision, recall, F1-Score, dan IoU, meskipun mAP tetap menunjukkan kinerja baik secara keseluruhan dalam berbagai kondisi pengujian.

DAFTAR PUSTAKA

[1] H. Hayat and H. Zayadi, "Model inovasi pengelolaan sampah rumah tangga," *JU-ke (Jurnal Ketahanan Pangan)*, vol. 2, no. 2, pp. 131–141, 2018.
[2] R. D. Ramadhani, A. N. A. Thohari, C. Kartiko, A. Junaidi, T. G. Laksana, and N. A. S. Nugraha, "Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 312–318, 2021.
[3] T. P. Sari, A. M. I. T. Asfar, A. M. I. A. Asfar, A. I. E. Rahayu, and A. S. N. Azizah, "Pemanfaatan Limbah Elektronik (E-Waste) Mix Resin pada Kelompok Karang Taruna Desa Batulappa," *E-Amal: Jurnal Pengabdian Kepada Masyarakat*, vol. 1, no. 3, pp. 491–496, 2021.
[4] R. D. Lisminingsih and A. Malikah, "Pendirian Unit Bank Sampah dan Pengelolaan Sampah Residu di Desa Parangargo Malang," in *Prosiding Seminar Nasional LPPM UMP*, 2019, pp. 148–157.
[5] R. Akbari, "Analisis Timbulan dan Komposisi Sampah di Kawasan Bhumi Merapi dan Stonehenge Kaliurang, Sleman, DI Yogyakarta," 2018.
[6] Sistem Informasi Pengelolaan Sampah Nasional (SIPSN), "Kementerian Lingkungan Hidup dan Kehutanan Direktorat Jenderal Pengelolaan Sampah, Limbah dan B3 Direktorat Penanganan Sampah." Accessed: Nov. 18, 2023. [Online]. Available: https://sipsn.menlhk.go.id/sipsn/
[7] D. Putri, R. M. A. Kinasti, and E. Lestari, "Pemanfaatan Limbah Abu Sisa Pembakaran Sampah Non Organik Sebagai Material Pengganti Pasir Pada Bata Beton Pejal," *Konstruksia*, vol. 10, no. 1, pp. 39–50, 2019.
[8] H. M. B. Tama, H. Helmy, and R. A. Mulyono, "Rancang Bangun Alat Pemilah Sampah Plastik Berbasis Sensorik RGB (Red, Green, Blue) sebagai Langkah Moderenisasi Teknologi pada Proses Pemilahan Sampah Plastik," *Blantika: Multidisciplinary Journal*, vol. 1, no. 4, pp. 241–249, 2023.
[9] A. I. Rasidi, Y. A. H. Pasaribu, A. Ziqri, and F. D. Adhinata, "Klasifikasi Sampah Organik dan Non-Organik Menggunakan Convolutional Neural Network," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 1, pp. 142–149, 2022.
[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[11] T. Chowdhury, A. R. Sarker, A. A. S. Rasel, and S. H. Fahim, "Bangladeshi Vehicle Identification via YOLO v8-Based License Plate Detection," 2024.
[12] K. R. Tanjung, L. Liliana, and H. Juwiantho, "Klasifikasi Benda Organik dan Anorganik Dengan Metode YOLOv3 dan ResNet50," *Jurnal Infra*, vol. 10, no. 2, pp. 268–274, 2022.
[13] O. S. N. Utomo, F. Utaminingrum, and E. R. Widasari, "Implementasi YOLO versi 3 untuk Mengidentifikasi dan Mengklasifikasi Sampah Kantor berbasis NVIDIA Jetson Nano," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 6, pp. 2829–2834, 2022.
[14] F. R. Hendri and F. Utaminingrum, "Rancang Bangun Sistem Pengklasifikasi Jenis Sampah Organik dan Anorganik menggunakan metode You Only Look Once versi 3 berbasis Raspberry Pi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 7, pp. 3509–3514, 2022.
[15] H. Purwantoro, T. Mudzakir, and S. Lestari, "Penerapan Algoritma YoloV5 Dalam Pendeteksian Objek Merek Sampah Botol Plastik," *Scientific Student Journal for Information, Technology and Science*, vol. 5, no. 1, pp. 18–23, 2024.
[16] M. Taşyürek and C. Öztürk, "A fine-tuned YOLOv5 deep learning approach for real-time house number detection," *PeerJ Comput Sci*, vol. 9, 2023, doi: 10.7717/PEERJ-CS.1453.
[17] S. A. Putri, G. Ramadhan, Z. Alwildan, I. Irwan, and R. Afriansyah, "Perbandingan Kinerja Algoritma YOLO Dan RCNN Pada Deteksi Plat Nomor Kendaraan," *Jurnal Inovasi Teknologi Terapan*, vol. 1, no. 1, pp. 145–154, 2023.
[18] A. Asroni, G. Indrawan, and L. J. E. Dewi, "Implementasi Hirarki Dataset Dalam Membangun Model Language Aksara Bali Menggunakan Framework Tesseract OCR," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 6, no. 1, pp. 20–28, 2023.
[19] E. I. Haksoro and A. Setiawan, "Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network," *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, vol. 5, no. 2, pp. 81–91, 2021.
[20] L. Satya, M. R. D. Septian, M. W. Sarjono, M. Cahyanti, and E. R. Swedia, "Sistem Pendeteksi Plat Nomor Polisi Kendaraan Dengan Arsitektur YOLOv8," *Sebatik*, vol. 27, no. 2, pp. 753–761, 2023.
[21] A. A. RAHMAN, S. D. AGUSTIN, N. U. R. IBRAHIM, and N. O. R. C. KUMALASARI, "Perbandingan Algoritma YOLOv4 dan Scaled YOLOv4 untuk Deteksi Objek pada Citra Termal," *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, vol. 7, no. 1, pp. 61–71, 2022.
[22] M. Sayed and G. Brostow, "Improved handling of motion blur in online object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1706–1716.