

IMPLEMENTASI ALGORITMA YOLO V8 UNTUK OBJECT DETECTION JENIS KENDARAAN DI JALAN RAYA

SKRIPSI

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Informatika



disusun oleh

FAIZ DAFFA KURNIA

22.11.4627

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2025

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi transportasi merupakan salah satu aspek penting yang tak dapat dipisahkan dari kehidupan manusia. Dalam kehidupan sehari-hari berbagai jenis kendaraan menjadi alat transportasi yang paling sering digunakan masyarakat untuk memenuhi kebutuhan mobilitas, baik untuk perjalanan kerja, belajar, distribusi barang, dan aktivitas lainnya. Hal ini menyebabkan lalu lintas di jalan raya menjadi padat terutama di daerah perkotaan tertentu di negara kita, sehingga pengelolaan lalu lintas dan transportasi menjadi tantangan yang penting untuk menjaga keamanan, kelancaran, dan ketertiban di jalan raya.

Berdasarkan data yang diperoleh dari Dinas Perhubungan Daerah Istimewa Yogyakarta, selama masa libur Natal dan Tahun Baru 2024–2025 tercatat 2,7 juta pergerakan kendaraan masuk-keluar DIY melalui kamera pantau pada empat titik utama yaitu Prambanan, Wates, Tempel, dan Piyungan[1]. Selain itu pada saat Lebaran April 2025, tercatat hingga 2,3 juta kendaraan masuk DIY dalam beberapa hari[2]. Hal ini menunjukkan bahwa volume kendaraan di Yogyakarta saja sangatlah tinggi dan dapat mengganggu aktivitas sehari-hari.

Salah satu langkah yang dapat mendukung pengelolaan lalu lintas adalah kemampuan untuk mendeteksi dan mengenali jenis kendaraan yang melintas secara otomatis. Dengan mengenali kendaraan secara akurat, dapat diperoleh data penting, seperti kepadatan lalu lintas, distribusi kendaraan berdasarkan jenis, dan pola pergerakan kendaraan. Informasi ini berguna untuk perencanaan, pengambilan keputusan, dan implementasi teknologi smart city.

Perkembangan teknologi *computer vision* dan *artificial intelligence*, khususnya *deep learning* terbukti efektif dalam mendeteksi sistem pemantauan lalu lintas dan mengklasifikasi kendaraan secara otomatis berdasarkan karakteristik visualnya dengan hasil yang akurat berdasarkan oleh penelitian terdahulu[3]. Hal ini otomatis membuka peluang untuk mendeteksi dan mengenali kendaraan secara real-time dari sebuah gambar atau video. Dengan menggunakan algoritma *object detection*, kendaraan yang melintas dapat dideteksi berdasarkan ciri-ciri visualnya, lalu dikelompokkan ke dalam beberapa jenis *class* yaitu motor, mobil, bus, truk, dan ambulans.

YOLO (*You Only Look Once*) merupakan salah satu algoritma *object detection* yang paling unggul dan populer digunakan saat ini karena proses deteksinya yang cepat dan akurat, YOLO dapat menyelesaikan prosesnya dengan sangat cepat karena hanya memerlukan sekali jalan saja dalam pemrosesannya, inilah yang membuatnya unggul dibandingkan algoritma yang lainnya[4]. Dalam perkembangannya, YOLO terus disempurnakan, dan saat ini tersedia versi terbaru, yaitu YOLOv8, yang mampu mendeteksi objek secara real-time, lebih akurat, dan lebih efisien[5]. Hal ini membuat YOLOv8 cocok diterapkan pada masalah deteksi kendaraan di jalan raya.

Berdasarkan latar belakang tersebut, pada penelitian ini akan diterapkan algoritma YOLOv8 untuk mendeteksi jenis kendaraan yang melintas di jalan raya. Sistem yang nantinya dibangun diharapkan dapat berguna sebagai alat bantu untuk pengelolaan lalu lintas, pengumpulan data kendaraan, dan perancangan teknologi lalu lintas pintar.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka rumusan masalah dalam penelitian ini adalah :

1. Bagaimana performa algoritma YOLOv8 untuk mendeteksi dan mengenali jenis kendaraan yang melintas di jalan raya dibandingkan metode lainnya?
2. Seberapa akurat algoritma YOLOv8 dapat mendeteksi dan mengklasifikasikan kendaraan berdasarkan jenisnya?

1.3 Batasan Masalah

Agar pembahasan pada penelitian lebih terarah dan rinci, maka masalah dibatasi pada:

1. Penelitian difokuskan pada deteksi dan pengenalan 5 jenis kendaraan, yaitu mobil, motor, bus, truk dan ambulan.
2. Sistem yang dibangun menggunakan algoritma deteksi YOLOv8.
3. Pengujian dan implementasi menggunakan citra dan video yang diambil pada kondisi lalu lintas normal pada siang hari dan cuaca cerah.
4. Penelitian tidak mencakup perbedaan merk, nomor plat, atau ukuran kendaraan.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan algoritma YOLOv8 pada sebuah sistem untuk mendeteksi dan mengenali jenis kendaraan di jalan raya.
2. Menguji dan menganalisa kinerja algoritma YOLOv8 berdasarkan akurasi deteksi dan klasifikasi kendaraan.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini antara lain:

1. Memberikan kemudahan pengumpulan data lalu lintas bagi pengelola lalu lintas.
2. Sebagai salah satu sarana pendukung terciptanya lalu lintas yang lebih teratur, aman, dan efisien, demi kenyamanan dan keselamatan pengguna jalan.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini terdiri dari lima bab, yang masing-masing dijelaskan sebagai berikut:

1. **BAB I PENDAHULUAN**

Berisi latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. **BAB II TINJAUAN PUSTAKA**

Berisi studi literatur dari penelitian terdahulu, dan dasar-dasar teori yang digunakan dalam penelitian ini.

3. **BAB III METODE PENELITIAN**

Menjelaskan objek penelitian, alur penelitian, metode yang digunakan, alat dan bahan, serta tahapan implementasi.

4. **BAB IV HASIL DAN PEMBAHASAN**

Berisi hasil implementasi sistem dan pengujian, serta pembahasan dari hasil yang didapat.

5. **BAB V PENUTUP**

Memuat kesimpulan akhir dari hasil penelitian dan saran untuk pengembangan lebih lanjut.

BAB II TINJAUAN PUSTAKA

2.1 Studi Literatur

Dalam penyusunan penelitian ini, ada beberapa referensi yang menjadi patokan dari penelitian yang dilakukan yang berkaitan dengan latar belakang masalah, penelitian yang berhubungan dengan object detection untuk mendeteksi kendaraan di jalan raya menggunakan algoritma YOLO v8 telah berkembang dalam beberapa tahun terakhir. Berikut ini adalah penelitian dari beberapa studi yang sesuai :

Penelitian berjudul “*Object Tracking Menggunakan Algoritma YOLOv8 untuk Menghitung Kendaraan.*” oleh Nurhaliza Juliyani Hayati et al mengembangkan metode pelacakan kendaraan menggunakan algoritma YOLOv8 dan DeepSORT. Fokus penelitian ini adalah pada kendaraan yang melakukan pelacakan secara real-time, dengan hasil akurasi, precision, dan recall mencapai 86%. Penelitian ini menyoroti masalah kemacetan lalu lintas yang semakin sulit diatasi dan membandingkan efektivitas deteksi objek statis serta klasifikasi jenis kendaraan[6].

Sementara itu, Iwan Virgawana et al. (2023) dengan judul penelitian “*Deteksi dan Tracking Objek Real-Time menggunakan YOLOv3.*” meneliti deteksi dan pelacakan objek menggunakan YOLOv3, dengan pendekatan evaluasi pada variasi pencahayaan dan posisi kamera pada jarak 20–100 cm. Hasilnya menunjukkan bahwa kendala utama adalah akurasi deteksi di lingkungan nyata yang cenderung bervariasi[7].

Penelitian “*Klasifikasi Jenis Kendaraan Menggunakan CNN dan Image Processing.*” oleh A. Ramadhani et al. (2022) mengkaji klasifikasi jenis kendaraan dengan pendekatan CNN dan *Image Processing* tanpa *bounding box*. Meskipun metode yang digunakan lebih sederhana, penelitian ini mencapai akurasi hingga 87%, dan menyoroti kebutuhan sistem monitoring lalu lintas untuk klasifikasi kendaraan berbasis teknologi visual[8].

Y. Pratama et al. (2023) dalam penelitiannya “*Penerapan Faster R-CNN untuk Deteksi Mobil di Jalan Raya.*” menerapkan algoritma *Faster R-CNN (region proposal)* dan VGG16 untuk mendeteksi mobil di jalan raya. Hasil deteksinya mencapai 91%, meskipun waktu inferensi dinilai masih lambat. Penelitian ini menekankan kekurangan algoritma *Faster R-CNN* dibandingkan YOLOv8 dalam hal kecepatan dan akurasi untuk sistem deteksi *real-time*[9].

Selanjutnya, penelitian “*Analisis Deteksi Citra Senjata Menggunakan YOLOv8.*” oleh Imam Maulana et al. (2023) menggunakan YOLOv8 untuk analisis deteksi citra senjata. Hasilnya menunjukkan mAP sebesar 84%, meskipun diuji pada kondisi gambar yang buram, gelap, dan bising. Penelitian ini menggarisbawahi meningkatnya kebutuhan terhadap sistem keamanan otomatis untuk mendeteksi objek berbahaya[10].

Adapun perbedaan penelitian saya dari jurnal-jurnal tersebut adalah fokus penelitian ini pada pengembangan sistem deteksi multi-class kendaraan di jalan raya secara real-time menggunakan algoritma YOLOv8, yang diarahkan untuk mendukung kebutuhan monitoring lalu lintas berbasis visual dan otomatisasi dengan akurasi tinggi.

Tabel 2.1 Keaslian Penelitian

| No | Judul penelitian | Nama Penulis | Tahun Publikasi | Hasil Penelitian | Perbandingan Penelitian | Problem | Metode |
|----|--|---------------------------------|-----------------|---|---|--|------------------------|
| 1 | Object Tracking Menggunakan Algoritma YOLOv8 untuk Menghitung Kendaraan. | Nurhaliza Juliyani Hayati et al | 2023 | Fokus pada kendaraan dengan melakukan tracking, Hasil akurasi, precision, dan recall mencapai 86%. | Fokus pada deteksi objek statis dan multiclass jenis kendaraan | Banyaknya kasus kemacetan dalam lalu lintas yang semakin susah untuk dihadapi . | YOLO v8 + DeepSORT |
| 2 | Deteksi dan Tracking Objek Real-Time menggunakan YOLOv3. | Iwan Virgiawan et al | 2023 | Deteksi objek dengan algoritma YOLO v3 dengan prinsip evaluasi variasi pencahayaan dan posisi kamera dengan jarak 20 - 100cm. | Perbedaan pada objek penelitian dan algoritma yang digunakan YOLO v3 vs YOLO v8 | Kendala akurasi deteksi objek di lingkungan nyata yang memiliki pencahayaan dan posisi kamera bervariasi | YOLO v3 |
| 3 | Klasifikasi Jenis Kendaraan Menggunakan CNN dan Image Processing. | A. Ramadhani et al | 2022 | Menggunakan metode CNN dan Image Processing sederhana tanpa bounding box untuk klasifikasi jenis kendaraan. Hasil akurasi mencapai 87%. | Jenis algoritma berbeda, tujuannya deteksi objek, berbeda dengan klasifikasi. | Kebutuhan sistem monitoring lalu lintas berbasis teknologi untuk klasifikasi kendaraan . | CNN + Image Processing |

| | | | | | | | |
|---|---|--------------------|------|---|---|--|--------------|
| 4 | Penerapan Faster R-CNN untuk Deteksi Mobil di Jalan Raya. | Y. Pratama et al | 2023 | Menggunakan algoritma Faster R-CNN (region proposal) dan VGG16. Hasil deteksi mencapai 91%. Waktu inferensi lebih lambat. | Jenis algoritma yang saya gunakan adalah YOLO v8 yang lebih baru dan akurat | Kurangnya sistem deteksi jenis kendaraan yang bersifat real time untuk pengawasan lalu lintas | Faster R-CNN |
| 5 | Analisis Deteksi Citra Senjata Menggunakan YOLOv8 | Imam Maulana et al | 2023 | Menggunakan algoritma YOLOv8, mAP 84%. Menguji pada kondisi gambar yang blur, gelap, dan terotasi. | Menggunakan algoritma yang sama YOLO v8 namun objek penelitian nya berbeda | Meningkatnya kebutuhan keamanan otomatis untuk mendeteksi objek berbahaya dalam berbagai kondisi citra | YOLO v8 |

2.2 Dasar Teori

2.2.1 Deep Learning

Deep Learning adalah salah satu dari sekian banyak cabang machine learning yang berjalan dengan menggunakan artificial neural networks. Algoritma ini dirancang khususnya untuk mencontoh cara berpikir manusia nyata dan memiliki kemampuan belajar dari kumpulan data besar. Hal tersebut dapat dilakukan karena deep learning memungkinkan model untuk secara otomatis mengekstrak fitur dari data mentah, terutama data berbentuk gambar, suara, dan juga teks[11].

2.2.2 Object Detection

Object detection merupakan teknik untuk mengidentifikasi dan menemukan objek tertentu dalam sebuah gambar ataupun video menggunakan neural networks. Object detection juga memberikan koordinat dari bounding box yang membatasi setiap objek yang terdeteksi[12]. Teknologi ini merupakan salah

satu bentuk dari kecerdasan buatan dengan cara training pada model agar dapat belajar layaknya belajarnya manusia, terutama dalam mendeteksi objek dalam kategori atau *class* tertentu.

2.2.3 Algoritma *You Only Look Once* (YOLO)

YOLO (*You Only Look Once*) adalah algoritma deteksi objek berbasis *deep learning* dengan pendekatan *single-stage*, yang memungkinkan deteksi dilakukan dalam satu proses prediksi. Algoritma ini dirancang untuk melakukan deteksi object secara real time karena kecepatan dalam pemrosesannya yang sangat tinggi. YOLO bekerja dengan membagi input suatu gambar ke dalam grid, kemudian secara bersamaan memprediksi bounding box serta class objek untuk setiap bagian dari grid tersebut dalam satu kali proses[4].

2.2.4 YOLO v8

Sejak pertama kali diperkenalkan, YOLO telah mengalami berbagai pembaruan versi, mulai dari YOLOv1 hingga YOLOv7 yang pada perkembangan tiap versinya berfokus pada peningkatan akurasi, efisiensi model, dan kemampuan generalisasi terhadap berbagai jenis objek dan kondisi gambar. Versi terbaru YOLO saat ini adalah YOLOv8 yang dikembangkan oleh Ultralytics, versi v8 ini memberikan peningkatan dari sisi akurasi, fleksibilitas model, dan dukungan framework modern dibandingkan dengan versi sebelumnya. Selain itu, YOLOv8 ditulis ulang menggunakan framework PyTorch murni dan tidak lagi menggunakan file konfigurasi .cfg, sehingga integrasi dan modifikasi lebih mudah dilakukan[5].

Arsitektur YOLOv8 terdiri 3 bagian yaitu *Backbone*, *Neck*, dan *Head*. *Backbone* digunakan untuk mengekstraksi fitur dari gambar input. YOLOv8 menggunakan varian *backbone* yang disebut C2f (*Cross-Stage Partial connections v2*) yang menggabungkan efisiensi dan kedalaman jaringan. *Backbone* ini mirip dengan CSP pada YOLOv5, namun dioptimalkan untuk kinerja yang lebih baik.

Bagian *neck* bertugas menggabungkan fitur dari berbagai tingkat (*multi-scale feature aggregation*) untuk meningkatkan akurasi deteksi pada objek kecil dan besar. YOLOv8 menggunakan arsitektur FPN (*Feature Pyramid Network*) yang dikombinasikan dengan PAN (*Path Aggregation Network*) untuk menyempurnakan representasi fitur.

Head bertugas menghasilkan prediksi akhir berupa *bounding box*, *confidence score*, dan *class probabilities*. Pada YOLOv8, head ini sudah mendukung *anchor-free detection*, yang berarti tidak bergantung pada *anchor box* seperti versi sebelumnya. Ini membuat proses pelatihan lebih sederhana dan model menjadi lebih ringan[13].

BAB III

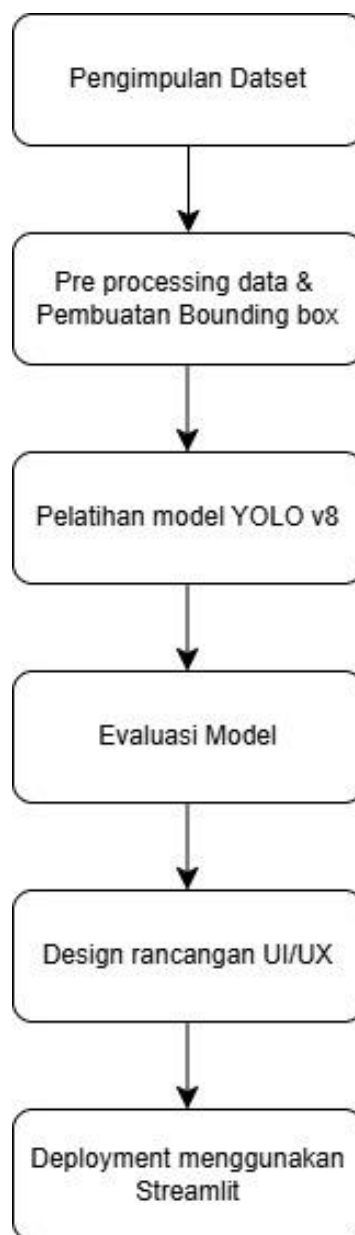
METODE PENELITIAN

3.1 Objek Penelitian

Objek Penelitian yang digunakan dalam penelitian ini adalah dataset berupa gambar statis jenis jenis kendaraan yang melintas di jalan raya yang nantinya akan diproses melalui model YOLO v8. Dataset diperoleh melalui Kaggle, salah satu situs sumber dataset public paling terpercaya saat ini.

3.2 Alur Penelitian

Alur penelitian yang dilakukan untuk pengembangan sistem ini adalah sebagai berikut:



1. Pemilihan dan pengumpulan dataset
Mengambil dataset berupa gambar kendaraan dari sumber publik Kaggle. Dataset berisi gambar dengan format jpg dan disertai label.
2. Pre processing data & Bounding box
Sebelum pengolahan data, langkah yang dilakukan adalah pre processing data untuk memastikan data yang digunakan dalam pelatihan model baik dan sesuai. Data juga dilengkapi dengan bounding box untuk digunakan saat pelatihan nanti.
3. Training model YOLO V8
Setelah melakukan pre processing data, Data akan dilatih dengan model YOLO V8 menggunakan library Ultralytics. Di tahapan ini perlu menentukan parameter tuning yang tepat untuk mendapatkan hasil model yang maksimal.
4. Evaluasi Model
Setelah pelatihan, model dievaluasi menggunakan metrik mAP (mean Average Precision), precision, dan recall terhadap dataset validasi untuk menentukan seberapa akurat model yang dilatih.
5. Perancangan UI/UX Sistem
Menentukan rancangan desain Interface menggunakan Figma.
6. Implementasi sistem
Model yang sudah dilatih diimplementasikan ke aplikasi Streamlit melalui Github, sehingga dapat diakses langsung oleh pengguna.

3.3 Alat dan Bahan

3.3.1 Data Penelitian

Dataset yang digunakan dalam penelitian ini berupa kumpulan gambar statis kendaraan yang melintas di jalan raya. Dalam pemilihan dataset, saya memutuskan untuk mengambil sumber dari Kaggle. Dataset yang diambil berjudul "*VehicleDetection-YOLOv8 - Detect cars, bikes, ambulances, trucks and motorcycles*" yang dipublish oleh Alkan Erturan.

Isi dataset ini berjumlah 2510 file, yang didalamnya berisi 1254 gambar dengan format .jpg, lalu 1254 label.txt yang sesuai dengan tiap gambarnya masing, masing, 1 buah file konfigurasi dataset.yaml, dan 1 buah Test Video yang di penelitian ini tidak saya gunakan. Dataset memiliki 5 jenis class utama : Ambulance, Bus, Car, Motorcycle, Truck.

3.3.2 Alat/Instrumen penelitian

Perangkat yang digunakan dalam proses pengembangan sistem ini adalah sebagai berikut :

1. Perangkat hardware

- Processor Intel® Core™ i7-4790
- GPU Radeon RX580 2048sp
- RAM 32GB
- Storage SSD 500GB
- OS Windows 11

2. Python 3.13

Bahasa pemrograman utama yang digunakan dalam pengembangan sistem ini. Python memiliki berbagai library yang sangat luas, terutama library terpenting dalam penelitian yaitu Ultralytics untuk menjalankan model YOLO v8.

3. Google Colab

Platform cloud yang digunakan untuk melakukan proses pelatihan model YOLOv8 dengan memanfaatkan GPU gratis yaitu Tesla T4 yang disediakan oleh Google.

4. Visual Studio Code

Text editor yang digunakan untuk menulis, mengelola, dan menjalankan kode Python selama pembuatan sistem.

5. Github

Platform version control berbasis Git yang digunakan untuk menyimpan, membackup, serta membagikan kode program secara online dan kolaboratif. Selain itu github juga digunakan sebagai platform penghubung ke streamlit cloud untuk melakukan deploy website.

6. Streamlit

Merupakan framework berbasis Python yang digunakan untuk membangun antarmuka web interaktif dari sistem deteksi kendaraan. Dengan Streamlit, sistem dapat diakses melalui browser secara langsung oleh pengguna tanpa perlu instalasi lokal.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Metode Pengujian

Pengujian dilakukan pada dataset validasi yang berisi gambar kendaraan yang berbeda dengan data pelatihan. Model diuji menggunakan data dari folder valid/labels dengan total 250 gambar uji. Adapun Parameter evaluasi yang digunakan untuk mengukur kinerja model deteksi kendaraan dalam penelitian ini adalah sebagai berikut:

1. **Precision**

Precision adalah tingkat **ketepatan model dalam mendeteksi kendaraan**, atau proporsi deteksi yang benar dari seluruh hasil deteksi yang dilakukan oleh model. Semakin tinggi nilai precision, semakin sedikit kesalahan deteksi (*false positives*). Precision yang tinggi menunjukkan bahwa model hanya menampilkan prediksi yang benar-benar akurat.

2. **Recall**

Recall adalah tingkat **kemampuan model dalam mendeteksi semua kendaraan yang ada** pada citra atau video. Semakin tinggi nilai recall, semakin banyak kendaraan yang berhasil dideteksi tanpa terlewatkan (*false negatives*).

3. **mAP50**

mAP50 adalah rata-rata akurasi deteksi pada seluruh kelas kendaraan, dihitung berdasarkan nilai **Average Precision (AP)** pada **threshold Intersection over Union (IoU) sebesar 50%**. IoU adalah perbandingan antara area **bounding box hasil deteksi** dan area **bounding box ground truth**. IoU 50% berarti model dianggap benar jika prediksi bounding box **menutupi setidaknya 50% area ground truth**.

Nilai mAP50 yang tinggi menunjukkan bahwa sebagian besar deteksi bounding box mendekati posisi yang benar dengan cukup baik.

4. **mAP50-95**

mAP50-95 adalah **metrik evaluasi yang lebih ketat dan komprehensif**, yang menghitung rata-rata AP pada beberapa threshold IoU mulai dari **50% hingga 95%** dengan interval 5%. Semakin tinggi mAP50-95, berarti model tidak hanya mendeteksi kendaraan dengan baik tetapi juga menghasilkan **bounding box yang presisi dan konsisten** di berbagai tingkat ketelitian. mAP50-95 merupakan **standar evaluasi resmi dalam COCO dataset**, yang mencerminkan kualitas deteksi dalam berbagai skenario.

4.2 Hasil Pengujian

```
Ultralytics 8.3.124 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,006,623 parameters, 0 gradients, 8.1 GFLOPs
val: Fast image access (ping: 12.6±27.1 ms, read: 10.4±11.1 MB/s, size: 31.4 KB)
val: Scanning /content/drive/MyDrive/Semester 6/PDM/VehiclesDataset/valid/labels.cache... 250 images, 0 backgrounds, 0 corrupt: 100% | 250/250 [00:00<?, ?it/s]
Class      Images  Instances  Box(P  R  mAP50  mAP50-95): 100% | 16/16 [00:03:00:00, 4.57it/s]
all         250      454      0.679  0.553  0.592  0.45
Ambulance   50       64      0.726  0.844  0.817  0.713
Bus         30       46      0.638  0.614  0.663  0.567
Car         90      238      0.561  0.472  0.5   0.34
Motorcycle  42       46      0.677  0.456  0.536  0.291
Truck       38       60      0.79   0.477  0.446  0.338
Speed: 1.2ms preprocess, 3.0ms inference, 0.0ms loss, 2.1ms postprocess per image
Results saved to /content/drive/MyDrive/Semester 6/PDM/VehiclesDataset/yolo_training_output/yolov8_vehicles_model2
ultralytics.utils.metrics.DetMetrics object with attributes:
```

| Class | Images | Instances | Precision | Recall | mAP50 | mAP50-95 |
|-------------|--------|-----------|-----------|--------|-------|----------|
| Ambulance | 50 | 64 | 0.726 | 0.844 | 0.817 | 0.713 |
| Bus | 30 | 46 | 0.638 | 0.614 | 0.663 | 0.567 |
| Car | 90 | 232 | 0.561 | 0.472 | 0.617 | 0.529 |
| Motorcycle | 42 | 46 | 0.677 | 0.456 | 0.536 | 0.291 |
| Truck | 38 | 60 | 0.79 | 0.477 | 0.446 | 0.338 |
| All Classes | 250 | 454 | 0.679 | 0.553 | 0.592 | 0.450 |

Secara keseluruhan, model memperoleh mAP50 sebesar **59,2%** dan mAP50-95 sebesar **45%**, yang menunjukkan performa yang cukup baik pada data validasi.

4.3 Pembahasan

Hasil pengujian model object detection pada penelitian ini menunjukkan performa yang cukup baik dalam mendeteksi beberapa jenis kendaraan dengan mAP50 mencapai **59,2%** dan mAP50-95 sebesar **45%**. Hal ini menunjukkan bahwa algoritma YOLOv8 mampu diterapkan untuk mendeteksi kendaraan secara otomatis, meskipun masih terdapat perbedaan akurasi antar kelas kendaraan.

Jika dibandingkan dengan penelitian yang dilakukan oleh Nurhaliza Juliyani Hayati et al. (2023) dalam jurnal KOMPUTA, terdapat beberapa perbedaan:

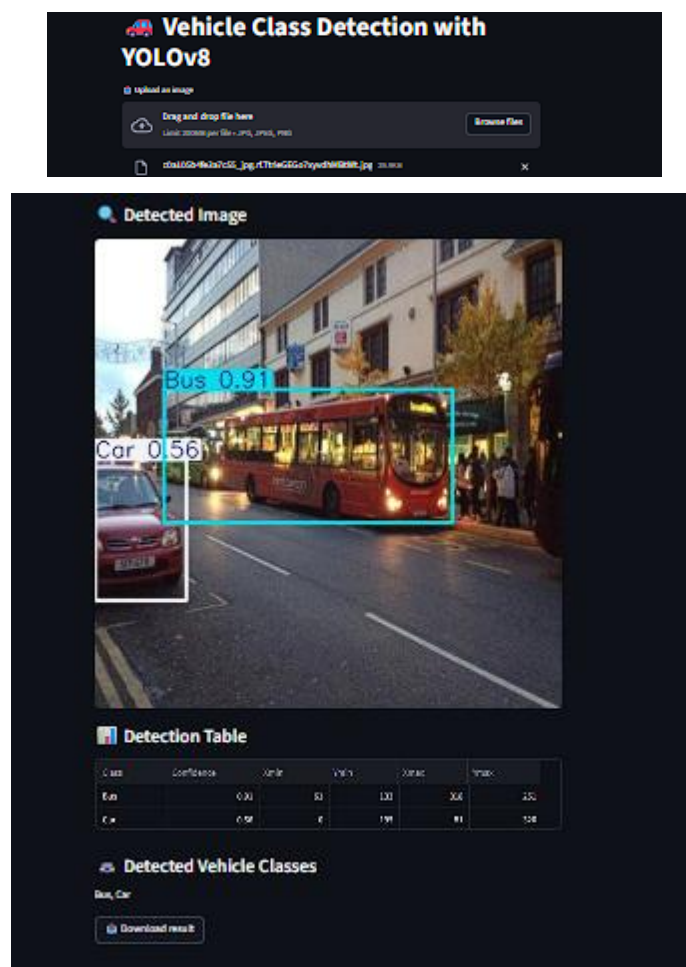
1. Penelitian Hayati et al. menggunakan algoritma YOLOv8 yang dikombinasikan dengan DeepSORT untuk melakukan *tracking* kendaraan dalam video, dengan fokus pada penghitungan jumlah kendaraan yang melintas.
2. Sedangkan dalam penelitian ini, sistem hanya melakukan deteksi objek statis dalam satu frame, bukan tracking, tetapi memiliki cakupan yang lebih luas karena mengklasifikasikan kendaraan ke dalam beberapa jenis, seperti car, bus, truck,

motorcycle, dan ambulance.

3. Dari sisi hasil, Hayati et al. berhasil mencapai akurasi 86%, sedangkan penelitian ini memiliki performa bervariasi antar kelas kendaraan dengan mAP50 tertinggi pada kelas Ambulance (81,7%) dan yang terendah pada Motorcycle (53,6%).
4. Penelitian Hayati et al. lebih menekankan pada implementasi penghitungan kendaraan pada jalur lalu lintas, sedangkan penelitian ini lebih fokus pada klasifikasi jenis kendaraan sebagai data tambahan dalam manajemen lalu lintas pintar (*smart traffic management*).

4.4 Hasil Implementasi Sistem

Hasil dari penelitian ini adalah sebuah sistem object detection jenis kendaraan menggunakan algoritma YOLOv8 dan diimplementasikan ke dalam aplikasi berbasis web menggunakan Streamlit. Sistem yang dibangun mampu mendeteksi dan mengenali beberapa jenis kendaraan pada gambar atau video jalan raya, serta menampilkan hasil deteksi dalam bentuk bounding box dan confidence score.



Vehicle Class Detection with YOLOv8

Upload an image

Drag and drop file here

Local sources: jpg, jpeg, png

Screen shot

selectedVehicle.jpg 20.00 KB

Detected Image

Bus 0.91

Car 0.56

Detection Table

| Class | Confidence | xmin | ymin | xmax | ymax |
|-------|------------|------|------|------|------|
| Bus | 0.91 | 83 | 103 | 306 | 250 |
| Car | 0.56 | 0 | 198 | 81 | 108 |

Detected Vehicle Classes

Bus, Car

Download result

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal berikut:

1. Performa algoritma YOLOv8 dalam mendeteksi dan mengenali jenis kendaraan di jalan raya menunjukkan hasil yang cukup baik. Model mampu mendeteksi kendaraan seperti mobil, bus, truck, ambulance, dan motorcycle dengan Tingkat akurasi yang terbilang cukup. Meskipun demikian, akurasi deteksi untuk beberapa jenis kendaraan masih berada di bawah penelitian terdahulu yang menggunakan metode object tracking (YOLOv8 + DeepSORT), terutama karena penelitian ini hanya menggunakan deteksi single-frame tanpa tracking lanjutan.
2. Berdasarkan hasil pengujian, algoritma YOLOv8 memperoleh rata-rata mAP50 sebesar 59,2% dan mAP50-95 sebesar 45%. Kelas Ambulance memiliki akurasi terbaik, sedangkan kelas Motorcycle masih perlu ditingkatkan karena performa mAP-nya masih rendah. Hal ini menunjukkan bahwa YOLOv8 cukup akurat dalam mendeteksi kendaraan yang berukuran besar atau dengan visual yang lebih jelas, namun masih memiliki tantangan dalam mendeteksi objek kecil dan jauh seperti sepeda motor.

REFERENSI

- [1] Dinas Perhubungan DIY, "Total Pergerakan Kendaraan Capai 2,7 Juta Selama Libur Nataru," *krjogja.com*, 2 Jan 2024.
<https://www.krjogja.com/berita-lokal/diy/total-pergerakan-kendaraan-capai-27-juta-selama-libur-nataru/>
- [2] Tribun Jogja, "Dinas Perhubungan DIY: 2,3 Juta Kendaraan Masuk Yogyakarta Selama Libur Lebaran," *TribunJogja.com*, 14 Apr 2024.
<https://jogja.tribunnews.com/2024/04/14/dishub-diy-23-juta-kendaraan-masuk-yogyakarta-selama-libur-lebaran>
- [3] Zhang, C., Lu, C., Li, S., & Wu, J. (2021). Vehicle Detection Based on Deep Learning in the Context of Intelligent Transportation Systems. *IEEE Access*, 9, 70975–70985.
- [4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- [5] Jocher, G., Chaurasia, A., Qiu, J., & Stoken, A. (2023). YOLO by Ultralytics: Version 8 (YOLOv8).
<https://github.com/ultralytics/ultralytics>
- [6] N. J. Hayati, D. Singasatia, M. R. Muttaqin, T. Informatika, and S. T. Wastukencana, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan," *KOMPUTA: Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023.
<https://ojs.unikom.ac.id/index.php/komputa/article/view/10654>
- [7] I. Virgiawan, F. Maulana, M. A. Putra, D. D. Kurnia, and E. Sinduningrum, "Deteksi dan Tracking Objek Secara Real-Time Berbasis Computer Vision Menggunakan Metode YOLO V3," *Jurnal Ilmiah Multidisiplin Indonesia*, vol. 3, no. 3, 2023.
<https://journal.ikopin.ac.id/index.php/humantech/article/view/4348>
- [8] A. Ramadhani, I. Hidayat, and N. A. Aziz, "Klasifikasi Jenis Kendaraan Menggunakan CNN dan Image Processing," *Jurnal PIKSEL: Penelitian Ilmu Komputer Sistem Embedded dan Logic*, vol. 10, no. 2, pp. 45–52, 2022.
<https://jurnal.unismabekasi.ac.id/index.php/piksel/article/download/9953/3316/22039>
- [9] Y. Pratama, M. S. Fadhil, and R. Wicaksono, "Penerapan Faster R-CNN untuk Deteksi Mobil di Jalan Raya," *Jurnal Sistem Informasi*, vol. 9, no. 1, pp. 20–27, 2023.
<https://journal-isi.org/index.php/isi/article/view/912>

- [10] I. Maulana, N. Rahaningsih, and T. Suprpti, "Analisis Deteksi Citra Senjata Menggunakan YOLOv8," *KOMPUTA: Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023. <https://ojs.unikom.ac.id/index.php/komputa/article/view/10654>
- [11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
<https://doi.org/10.1038/nature14539>
- [12] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.
<https://doi.org/10.1109/TNNLS.2018.2876865>
- [13] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2117–2125.

LAMPIRAN

Link Dataset :

<https://www.kaggle.com/datasets/alkanerturan/vehicledetection>

<https://www.kaggle.com/datasets/zwartfreak/bike-images>

<https://app.roboflow.com/bikedataset-cdnxy/bike-dataset-yyopo/2>

Link colab :

https://colab.research.google.com/drive/1jgROkg_3aHZwfeS5sC5sncqV4wlvPU4

[https://colab.research.google.com/drive/1bZLle6W-](https://colab.research.google.com/drive/1bZLle6W-Lu24mnQuEVJOgOwm9PX_pQKd?usp=sharing)

[Lu24mnQuEVJOgOwm9PX_pQKd?usp=sharing](https://colab.research.google.com/drive/1bZLle6W-Lu24mnQuEVJOgOwm9PX_pQKd?usp=sharing)

Link Deploy Streamlit :

<https://app-app-dmcfcalzvd8rcewwm9bejs.streamlit.app/>