

OBJECT TRACKING MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO)v8 UNTUK MENGHITUNG KENDARAAN

Nurhaliza Juliyani Hayati¹, Dayan Singasatia², Muhamad Rafi Muttaqin³

^{1,2,3}Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana
Jl. Cikopak No. 53, Sadang, Purwakarta

E-mail : nurhalizajh@gmail.com¹, dayan@stt-wastukencana.ac.id², rafi@stt-wastukencana.ac.id³

Abstrak

Kendaraan adalah alat transportasi yang sudah ada pada zaman dahulu sampai saat ini, banyak masyarakat menggunakan kendaraan seperti mobil dan sepeda motor. Pencacahan jenis dan jumlah kendaraan dilakukan untuk mengumpulkan informasi data lalu lintas. Dalam memperoleh parameter data jumlah kendaraan, masih menggunakan perhitungan secara manual biasanya rentan dengan kesalahan serta memakan banyak waktu serta tenaga. Penerapan *Artificial Intelligence* seperti deteksi objek merupakan bidang pada *computer vision*. Pada *intelligent transportation system*, data lalu lintas menjadi kunci untuk melakukan penelitian serta merancang sebuah sistem. Untuk mengatasi permasalahan peneliti melakukan *object tracking* menggunakan algoritma *You Only Look Once (YOLO)v8* untuk mendeteksi jenis dan menghitung jumlah kendaraan. Metodologi yang diterapkan adalah *AI Project Cycle* tahapan yang digunakan *problem scoping*, *data acquisition*, *data exploration*, *modelling*, dan *evaluation confusion matrix*. Hasil *evaluation confusion matrix* diperoleh tingkat *accuracy* 89%, *precision* 89%, *recall* 90% dan perbandingan *precision*, *recall* yang dibobotkan diperoleh nilai *F1-Score* sebesar 89%. Dengan demikian algoritma *You Only Look Once (YOLO)v8* cukup akurat mendeteksi *object tracking* untuk menghitung kendaraan.

Kata kunci: *AI Project Cycle*, *Confusion Matrix*, *Computer Vision*, Kendaraan, *You Only Look Once (YOLO)v8*.

Abstract

Vehicles are a means of transportation that have existed from ancient times until now, many people use vehicles such as cars and motorbikes. Enumeration of types and numbers of vehicles is carried out to collect traffic data information. In obtaining data parameters for the number of vehicles, manual calculations are usually prone to errors and take a lot of time and energy. The application of Artificial Intelligence such as object detection is a field of computer vision. In intelligent transportation systems, traffic data is the key to conducting research and designing a system. To overcome the problem, researchers carried out object tracking using the You Only Look Once (YOLO) v8 algorithm to detect the type and count the number of vehicles. The methodology applied is the AI Project Cycle stages which use problem scoping, data acquisition, data exploration, modeling, and confusion matrix evaluation. The results of the confusion matrix evaluation obtained an accuracy level of 89%, precision of 89%, recall of 90% and a weighted comparison of precision and recall obtained an F1-Score value of 89%. Thus, the You Only Look Once (YOLO) v8 algorithm is accurate enough to detect object tracking to calculate vehicles.

Keywords: *AI Project Cycle*, *Confusion Matrix*, *Computer Vision*, Vehicles, *You Only Look Once (YOLO)v8*.

1. PENDAHULUAN

Kendaraan adalah transportasi yang telah ada dari zaman dahulu sampai masa kini, banyak masyarakat yang mengendarai kendaraan menggunakan mobil ataupun sepeda motor. Berdasarkan informasi ada tahun 2019 jumlah kendaraan di Indonesia kurang lebih 133,6 Juta Unit [1]. Peningkatan penggunaan transportasi setiap harinya selalu menyebabkan bertambahnya data kendaraan. Bertambahnya jumlah kendaraan tersebut memicu terjadi kemacetan pada beberapa wilayah di Indonesia [2].

Masalah ini mendapat beberapa perhatian kalangan sebagai ahli teknis, serta beberapa peneliti untuk menyelesaikan masalah kemacetan. Pencacahan jenis serta jumlah kendaraan dilakukan sebagai

upaya untuk mengetahui karakteristik lalu lintas, mengumpulkan informasi data lalu lintas serta untuk menilai kinerja lalu lintas [3]. Untuk mencegah adanya penumpukan jumlah kendaraan dilokasi tertentu diperlukan suatu data statistik perihal pertumbuhan kendaraan diwilayah tersebut tiap tahunnya. Dalam memperoleh kriteria jumlah kendaraan, pada umumnya masih sering memakai perhitungan manual biasanya rentan dengan kesalahan serta menguras waktu dan juga tenaga [4]. Berbeda seperti cara manual, sistem akan bisa melakukan perhitungan tanpa terikat waktu kerja dengan manusia, dikarenakan sudah dijalankan secara otomatis dan bisa mendeteksi citra kendaraan sehingga bisa diketahui jenis serta jumlahnya [5]. Seiring berkembangnya teknologi, berbagai ragam deteksi kendaraan sudah dibangun, salah satunya merupakan pendeteksian pada kendaraan sudah menggunakan algoritma seperti *deep learning* adalah pengembangan *neural network multiple layer* dengan memberi tugas untuk mendeteksi suatu objek dan sebagainya[6].

Penerapan *Artificial Intelligence* terus mengalami peningkatan, khususnya pada bidang *computer vision* yang memiliki tugas yang sama pada penggunaan komputer untuk mempelajari kepandaian manusia. Teknologi *computer vision* sudah banyak diterapkan pada berbagai bidang, termasuk keamanan, pekerjaan sosial, kesehatan, transportasi, serta bidang terkait lainnya. Pada *Intelligent Transportation System (ITS)*, data lalu lintas menjadi sebuah kunci untuk melakukan penelitian serta merancang sebuah sistem [7]. Tetapi tantangan sesungguhnya ada pada pengembangan metode yang cepat, efisien serta akurat secara komputasi untuk dipergunakan di dunia nyata [8].

Salah satu pengembangan terbaru untuk deteksi objek adalah dengan menggunakan *deep learning* yaitu *You Only Look Once (YOLO)*. YOLO merupakan pendekatan *object detection* dengan cara *real-time* menggunakan jaringan saraf *convolutional* [9], lapisan *convolutional* akan terjadi proses konvolusi pada setiap jaringan [10]. Pada penelitian ini menggunakan algoritma YOLO versi terbaru yakni versi YOLOv8, peneliti menggunakan algoritma ini untuk membuktikan fitur algoritma YOLOv8 yang baru saja di tawarkan. *State-Of-The-Art* adalah salah satu fitur utama YOLOv8, fitur ini dirancang menjadi kerangka kerja yang mendukung seluruh versi YOLO sebelumnya. Dengan menggunakan algoritma YOLOv8, diharapkan bisa memberikan hasil yang baik sebagai landasan atau acuan untuk terciptanya suatu sistem yang dapat membantu manajemen lalu lintas secara otomatis untuk mencapai perbaikan suatu instansi agar dapat meningkatkan pencapaian kinerjanya untuk mendeteksi serta menghitung jumlah kendaraan.

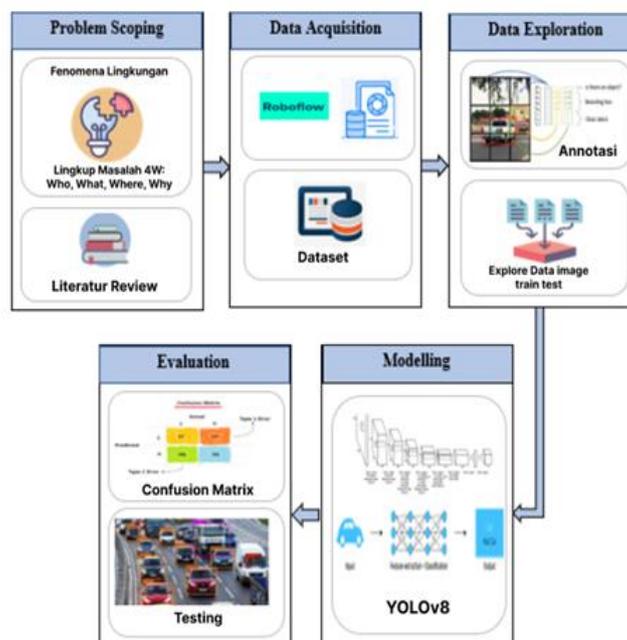
Roboflow merupakan *platform web* yang memiliki fungsi berhubungan dengan kumpulan dataset. *Roboflow* adalah kerangka kerja pengembang *computer vision* untuk pengumpulan data yang lebih baik hingga prapemrosesan, dan teknik pelatihan model. Dengan menggunakan *Roboflow* dapat membagikan dataset sekaligus memproses dataset tersebut melakukan annotate atau menandai objek yang akan di deteksi menggunakan *bounding box*, selain itu dapat digunakan juga pre-processing pada dataset misalnya melakukan *grayscale*, dan juga *augmentasi* dengan menggunakan *Roboflow*. Untuk melakukan pengujian deteksi objek diperlukan sebuah dataset pada tahap akuisisi data (pengumpulan data), dimana tahap tersebut merupakan sebuah tantangan bagi peneliti untuk mengumpulkan sekumpulan data citra kendaraan yang baik untuk melakukan pendeteksian objek.

Pada penelitian [11] menerapkan algoritma *You Only Look Once (YOLO)v3* yang bertujuan mendeteksi volume kendaraan dengan menggunakan dataset yang bersumber dari CCTV. Pengumpulan video berpengaruh oleh jarak kamera pada objek, sehingga menghasilkan pendeteksian objek yang masih kurang sesuai. Pada penelitian [12] mendeteksi kendaraan bermotor menggunakan video dari 3 algoritma yang digunakan. Pada algoritma YOLOv3 dapat diperoleh akurasi yang baik, tetapi sedikit lambat untuk kecepatan pergerakan akibat proses *learning* data dikarenakan terlalu lama. pada algoritma *Haar Cascade* diperoleh kecepatan yang baik, tetapi tingkat akurasi deteksi kurang baik dibandingkan dengan YOLOv3. Pada YOLOv3Tiny diperoleh kecepatan yang lebih baik dari YOLOv3, tetapi tingkat akurasi masih kurang dibandingkan YOLOv3 dan *Haar Cascade*. Pada penelitian [13] melakukan deteksi klasifikasi dan menghitung kendaraan menggunakan kamera CCTV dengan algoritma YOLOv3. Pengujian pendeteksian dilakukan dengan percobaan pengujian *training* dengan nilai *Precision* 99%, nilai *Recall* 90%, dan *F1 Score* sebesar 94%. Sedangkan hasil perhitungan deteksi kendaraan dengan mengambil video selama 1 menit dan 10 detik menghasilkan 48%.

Berdasarkan permasalahan yang sudah dijelaskan, pada penelitian ini bertujuan melakukan *object tracking* menggunakan algoritma *you only look once (YOLO)v8* yang dikembangkan oleh Ultralytics dengan mengevaluasi kinerja versi algoritma YOLOv8 berdasarkan keberhasilan dari arsitektur sebelumnya, membuat model yang terbaik untuk melakukan deteksi, klasifikasi, dan segmentasi objek secara real-time lebih cepat, lebih akurat, dan lebih sederhana dalam melakukan pendeteksian untuk menghitung jumlah kendaraan.

2. METODOLOGI

AI Project Cycle merupakan proses untuk pembuatan *project AI* secara terstruktur [14]. Dengan menerapkan metode *AI Project Cycle* yang dilakukan dengan beberapa tahapan, seperti ditunjukkan pada Gambar 1.



Gambar 1. Tahapan *AI Project Cycle*

2.1 Problem Scoping

Tahap penelitian awal melakukan observasi terhadap fenomena-fenomena lingkungan secara langsung, melakukan literatur review, kemudian melakukan identifikasi masalah untuk dijadikan bahan penelitian. Lingkup Masalah yang digunakan adalah 4W diantaranya *Who*, *What*, *Where* dan *Why*. Proses ini membantu mengidentifikasi serta menyelesaikan masalah dengan upaya yang lebih baik dan efisien.

2.2 Data Acquisition

Tahap akuisisi data dilakukan pengambilan dataset gambar kendaraan pada *website Roboflow* <https://universe.roboflow.com/> dengan memanfaatkan *Application Programming Interface (API)* yang merupakan sebuah antarmuka yang dapat menautkan sebuah aplikasi dengan aplikasi lainnya. API berperan menjadi perantara antar aplikasi, baik pada suatu *platform* yang sama ataupun dari lintas *platform* [15].

2.3 Data Exploration

Tahap *exploration* dikerjakan eksplorasi pada dataset yang sudah diperoleh pada tahap sebelumnya. Dilakukan anotasi data citra dianotasikan dengan membuat *bounding box* (kotak pembatas) pada objek dengan memberikan kelas yang sesuai pada setiap gambar [16], dataset yang sudah dianotasi kemudian akan dibagi menjadi *data testing*, *data validation* dan *data training*.

2.4 Modelling

Tahap *modelling* merupakan proses pelatihan algoritma sebagai tahap untuk *training phase* yang dilakukan untuk mendapatkan pola-pola data untuk melakukan prediksi serta membuat suatu keputusan. Proses konfigurasi program untuk melatih model yang dibuat menggunakan bahasa pemrograman python dengan menambahkan DeepSORT yang merupakan algoritma pelacakan *computer vision* untuk melacak objek sambil menetapkan ID ke setiap objek. Salah satu algoritma yang digunakan untuk *training* dengan menggunakan algoritma YOLOv8.

2.5 Evaluation

Tahap *evaluation* dilakukan sebagai pengkajian proses pemilihan model yang akan digunakan pada *project AI*. Proses evaluasi ini menggunakan *Confusion Matrix* dengan kombinasi berbeda dari nilai aktual

dan nilai prediksi [17], selanjutnya dilakukan *testing* menggunakan video dengan membuat garis *Region of Interest* (ROI) untuk mengidentifikasi suatu area sehingga mengoptimalkan suatu model untuk mendeteksi, menghitung serta mengklasifikasi kendaraan secara *realtime* [18], apakah model *pre-trained* yang dihasilkan sudah sesuai serta mengidentifikasi ketidaksesuaian hasil sebuah sistem dengan hasil yang diharapkan.

3. HASIL DAN PEMBAHASAN

3.1 Problem Scoping

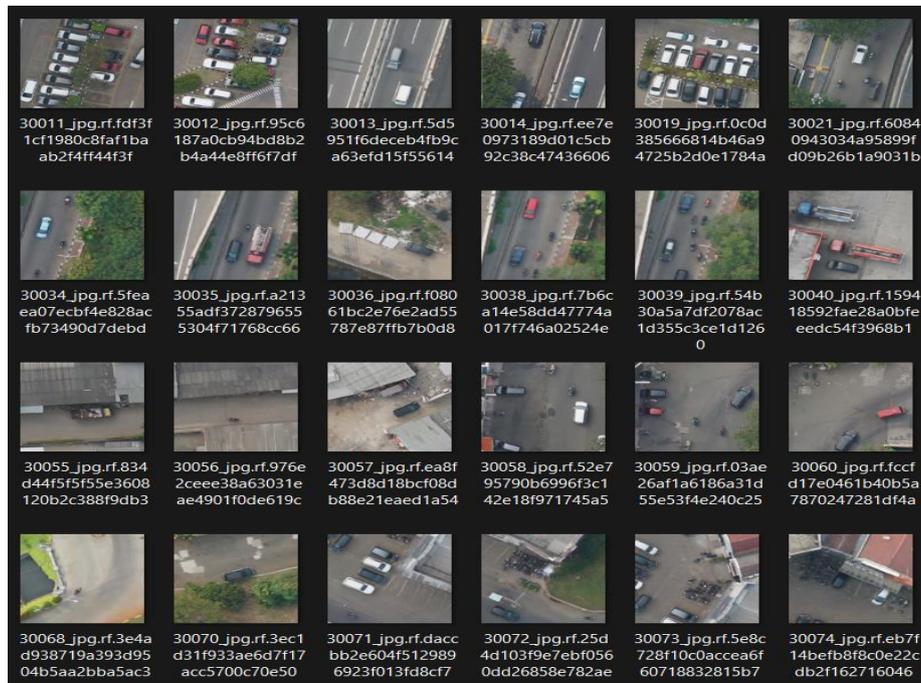
Tahap *Problem Scoping* dilakukan identifikasi masalah dilakukan pada 10 Mei 2023 melalui proses wawancara dengan Bapak Sumijan yang bertugas mengawasi lalu lintas di POS 5A Polres Purwakarta menggunakan 4W diantaranya adalah *Who, What, Where, dan Why* yang akan diteliti berdasarkan dari fenomena lingkungan. Hasil pada tahapan ini berupa siapa target, apa dan kapan masalah dapat terjadi serta solusi yang dapat diselesaikan untuk mengatasi masalah tersebut. Hasil dari tahapan ini dapat disimpulkan pada tabel *Problem Statement Template* seperti yang ditunjukkan pada Tabel 1.

Tabel 1. *Problem Statement Template*

Subjek:	Pihak kepolisian lalu lintas.	<i>Who</i>
Masalah berupa:	Mengumpulkan data lalu lintas dalam mendapatkan parameter jumlah kendaraan karena masih memprediksi data lalu lintas dengan perhitungan manual.	<i>What</i>
Pada Saat:	Di jalan raya.	<i>Where</i>
Solusi idealnya:	Menghitung jumlah kendaraan secara otomatis menggunakan algoritma YOLOv8.	<i>Why</i>

3.2 Data Acquisition

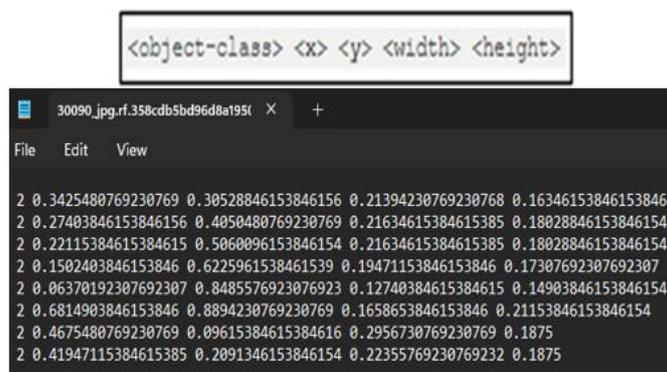
Proses pengumpulan dataset gambar kendaraan dilakukan pada *website Roboflow* dengan memanfaatkan *Application Programming Interface* (API) menggunakan *Google Colaboratory*. Dataset yang sudah dikumpulkan disimpan dengan format jpg pada sebuah folder yang menghasilkan dataset sebanyak 4.680 berikut hasil pengumpulan dataset citra kendaraan seperti ditunjukkan pada Gambar 2.



Gambar 2. Dataset kendaraan

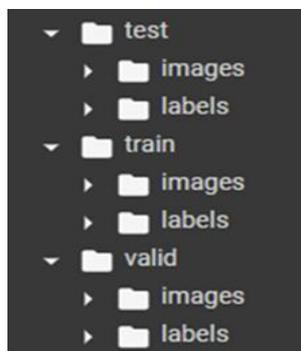
3.3 Data Exploration

Tahap *exploration* dilakukan eksplorasi terhadap dataset yang sudah didapatkan proses ini dijelaskan mengenai gambaran dari informasi data dengan mengeksplorasi dan mendeskripsikan struktur dataset yang sudah dilakukan dalam tahap pengumpulan data dan proses anotasi citra. Hasil dari proses akuisisi data mendapatkan dataset sebanyak 4680 merupakan kumpulan dataset *multiclass* yang sudah di anotasi yang terdiri dari 9 class pada website Roboflow. Proses anotasi pada setiap gambar dilakukan pelabelan dengan ukuran 416×416 , anotasi citra YOLO akan tersimpan menggunakan format file .txt dengan nama yang sama untuk setiap file gambar dalam direktori yang sama. Setiap file .txt berisi anotasi untuk file gambar terkait, yaitu kelas objek, koordinat objek, tinggi dan lebar contoh format anotasi citra YOLO pada file .txt seperti ditunjukkan pada Gambar 3.



Gambar 3. Hasil Anotasi

Hasil pada tahap eksplorasi struktur dataset yang sudah dianotasi akan terbagi menjadi *data testing*, *data validation* dan *data training* yang tersimpan pada sebuah folder dengan folder label seperti ditunjukkan pada Gambar 4.



Gambar 4. Folder Hasil Exploration

3.4 Modelling

Pada tahap *modelling* konfigurasi program dilakukan untuk melatih model yang dibuat menggunakan *Google Colaboratory* menggunakan bahasa pemrograman python. Algoritma yang digunakan untuk *training* adalah YOLOv8. Sebelum dilakukan *training* model dilakukan beberapa tahapan seperti *import libraries* dari *Ipython.display* untuk menampilkan gambar, melakukan *clone Github Repo* pada file *Google Colab Notebook*, menginstal beberapa *requirements* yang diperlukan, mengunduh file *DeepSORT* untuk menerapkan *object tracking* dengan menetapkan ID ke setiap objek, kemudian proses *training* model menggunakan algoritma YOLOv8 yang dilakukan sebanyak 50 iterasi berikut script yang digunakan dalam proses *training* model menggunakan *Google Colaboratory* seperti ditunjukkan pada Gambar 5.

```

- Training Model

[ ] HOME = '/content/YOLOv8-DeepSORT-Object-Tracking/ultralytics/yolo/v8/detect'

[ ] %cd {HOME}

/content/YOLOv8-DeepSORT-Object-Tracking/ultralytics/yolo/v8/detect

[ ] %cd {dataset.location}

/content/YOLOv8-DeepSORT-Object-Tracking/ultralytics/yolo/v8/detect/detection-1

[ ] %cd {HOME}

!python train.py model=yolov8.pt data={dataset.location}/data.yaml epochs=50 imgsz=640
    
```

Gambar 5. Proses Training

Hasil dari proses *training* akan menghasilkan *file weights* yang tersimpan pada *Google Drive*, berikut adalah *script* yang digunakan dalam proses *download file weights* pada *Google Drive* menggunakan *Google Colaboratory* seperti ditunjukkan pada Gambar 6.

```

- Downloading Weights pada Google Drive"

[19] %cd {HOME}

/content/YOLOv8-DeepSORT-Object-Tracking/ultralytics/yolo/v8/detect

[20] !gdown "https://drive.google.com/uc?id=1M61XXXaPpUEUOnpxIj03fjQRSSxmsSKf&confirm=t"

Downloading...
From: https://drive.google.com/uc?id=1M61XXXaPpUEUOnpxIj03fjQRSSxmsSKf&confirm=t
To: /content/YOLOv8-DeepSORT-Object-Tracking/ultralytics/yolo/v8/detect/best.pt
100% 368M/368M [00:11<00:00, 30.7MB/s]
    
```

Gambar 6. Download File Weights

3.5 Evaluation

Tahap *evaluation* dilakukan proses pemilihan model yang akan digunakan untuk hasil akhir. Berkaitan dengan data *training* yang sudah dilakukan pemodelan menggunakan algoritma *You Only Look Once* (YOLO)v8. Selanjutnya dilakukan pengujian evaluasi *confusion matrix* digunakan untuk membedakan hasil informasi yang digunakan oleh sistem dengan hasil informasi yang sebenarnya. Sebagai evaluasi maka perlu diukur pengujian *confusion matrix* yang terdiri dari akurasi (*accuracy*), ketepatan (*precision*) dan keberhasilan (*recall*). Berdasarkan pemodelan algoritma *You Only Look Once* (YOLO)v8 diperoleh hasil dalam bentuk *confusion matrix* dapat dilihat pada tabel 2.

Tabel 2. Confusion Matrix

Predicted Value	Actual Values						
	Camping Car	Car	Motorcycle	Pickup	Tractor	Truck	Van
Camping Car	71	0	0	0	0	0	0
Car	0	95	2	28	0	24	16
Motorcycle	0	0	88	0	0	0	0
Pickup	0	1	0	58	0	4	0
Tractor	0	0	0	0	67	0	0
Truck	5	0	0	0	0	56	0
Van	0	0	0	0	0	0	76

Pengukuran kinerja *confusion matrix* menggunakan 4 istilah yang akan menunjukkan hasil dari proses klasifikasi yaitu:

1. *True Positive* (TP) = Total dari data yang positif dan terklasifikasi oleh sistem dengan benar.
2. *True Negative* (TN) = Total dari data yang positif dan terklasifikasi oleh sistem dengan salah.
3. *False Positive* (FP) = Total dari data yang negatif dan terklasifikasi oleh sistem dengan benar.
4. *False Negative* (FN) = Total dari data yang negatif dan terklasifikasi oleh sistem dengan salah.

Accuracy adalah hasil akurasi untuk mendefinisikan tingkat dari dekatnya nilai aktual dengan nilai hasil prediksi. Semakin tinggi tingkat akurasi, semakin baik kinerja metode. Adapun rumus untuk melakukan perhitungan akurasi seperti pada rumus (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$= \frac{71+95+88+58+67+56+76}{591} = \frac{511}{591} = 0,86$$

Precision adalah tingkat keakuratan data yang diminta dengan data yang diberikan oleh model. *Precision* merupakan klasifikasi TP dari keseluruhan data yang diprediksi positif menjadi nilai kelas positif. Adapun rumus untuk melakukan perhitungan presisi seperti pada rumus (2).

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$= \frac{71}{71+0} + \frac{95}{95+70} + \frac{88}{88+0} + \frac{58}{58+5} + \frac{67}{67+0} + \frac{56}{56+5} + \frac{76}{76+0} = \frac{511}{591} = 0,86$$

Recall adalah tingkat dari keberhasilan dari sistem untuk mendapatkan informasi. Adapun rumus untuk melakukan perhitungan *recall* seperti pada rumus (3).

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$= \frac{71}{71+5} + \frac{95}{95+1} + \frac{88}{88+2} + \frac{58}{58+28} + \frac{67}{67+0} + \frac{56}{56+28} + \frac{76}{76+16} = \frac{511}{591} = 0,86$$

F1 Score yang merupakan perbandingan *precision* dan *recall*. Adapun rumus untuk melakukan perhitungan *F1 Score* seperti pada rumus (4).

$$F1\ Score = 2 \times \frac{precision \times Recall}{precision + Recall} \tag{4}$$

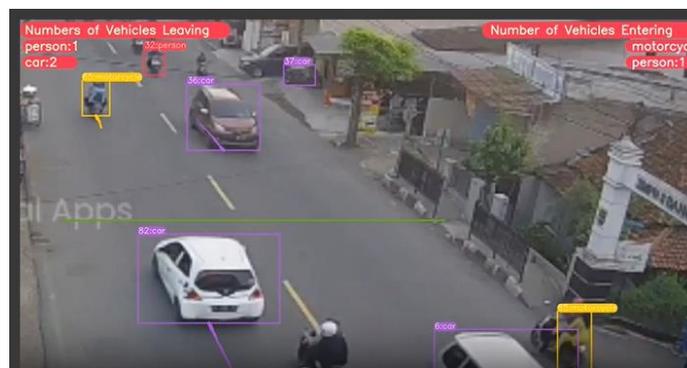
$$= 2 \times \frac{0,86 \times 0,86}{0,86 + 0,86} = 2 \times \frac{0,739}{1,72} = 0,85$$

Setelah melakukan proses *training* dan *evaluasi* model, dilakukan tahap *testing*. Berikut merupakan proses testing model pada demo video, dimana video yang akan di demo kan sudah di simpan pada *Google Drive* video yang didapatkan melalui aplikasi Papais CCTV dimana video tersebut berisi CCTV kendaraan yang berada di daerah purwakarta. Proses *testing object tracking* untuk menghitung jumlah kendaraan. Membuka file *predict.py* yang berisi *code* bahasa python untuk membuat garis roi yang akan dilewati untuk menghitung kendaraan dimana didalam file tersebut untuk menghitung jumlah kendaraan yang masuk dan menghitung jumlah kendaraan yang keluar serta menampilkan hasil jumlah kendaraan pada video akan muncul di atas pojok video. Hasil proses mengintegrasikan *object tracking* untuk menghitung jumlah kendaraan pada *Google Colab Notebook*, berikut adalah script yang digunakan dalam proses demo video pada *Google Colab Notebook* ditunjukkan pada Gambar 8.



Gambar 8. Testing 1

Berikut merupakan proses testing model pada demo video, dimana video yang akan di demo kan sudah di simpan pada Google Drive video yang didapatkan melalui aplikasi Papais CCTV dimana video tersebut berisi CCTV kendaraan yang berada di daerah purwakarta hasil demo video ditunjukkan pada Gambar 9.



Gambar 9. Testing 2

4. PENUTUP

Penelitian ini dilakukan untuk menguji algoritma *You Only Look Once* (YOLO)v8 dengan menambahkan *file DeepSORT* untuk menerapkan *object tracking* dengan menetapkan ID ke setiap objek untuk melakukan pendeteksian secara otomatis sehingga dapat mendeteksi kendaraan dengan mengetahui jenis dan jumlahnya. Proses penelitian menggunakan metodologi *AI Project Cycle* sampai dengan tahap *Evaluation*. Tahap *Data Acquisition* menggunakan data yang tersedia pada *website Roboflow* dengan memanfaatkan *Application Programming Interface* (API). Tahap *Evaluation* menggunakan *Confusion Matrix* diperoleh tingkat *accuracy* sebesar 86%, *precision* 86%, dan *recall* 86% sehingga ketepatan dan keberhasilan pendeteksian memiliki tingkat akurasi yang sama sebesar 86%, sedangkan perbandingan dari *precision* dan *recall* yang telah dibobotkan dengan diperoleh nilai *F1 Score* sebesar 85%. Hasil pengujian yang dilakukan pada penelitian ini, diperoleh hasil pendeteksian secara otomatis sehingga dapat mendeteksi kendaraan dengan mengetahui jenis dan jumlahnya mempunyai akurasi yang baik.

Untuk penelitian selanjutnya proses pengumpulan dataset dapat dilakukan dengan membuat dataset dengan *partnership* pihak ketiga atau dengan menggunakan open data seperti yang tersedia pada *Kaggle* ataupun *Github*. Serta dapat menggunakan Algoritma sebelum versi *You Only Look Once* (YOLO)v8 atau versi terbaru, untuk membandingkan performa pendeteksian *You Only Look Once* (YOLO) dan hasil *Confusion Matrix*.

DAFTAR PUSTAKA

- [1] S. C. Tirtawahjono and D. G. Purwanto, "Penentuan Distribusi Arus Lalu Lintas Pada Persimpangan Berbasis Teknologi Computer Vision & Deep Learning," *J. Dimens. Pratama Tek. Sipil*, vol. 11, no. 1, pp. 37–43, 2022.

- [2] Rubangiya, Harttati, and Y. Arie Wijaya, "Evaluasi Pembangunan Median Jalan," *Tekno. Transp. dan Logistik*, vol. 1, no. 1, pp. 11–14, 2020.
- [3] T. Harttati and Y. Arie Wijaya, "Analisis Data Lalu Lintas Jaringan di Kantor Canggih Cyber Operation Center Menggunakan Algoritma K-Means," *J. Ilm. NERO*, vol. 7, no. 1, p. 2022, 2022.
- [4] Y. Pratama and E. Rasywir, "Eksperimen Penerapan Sistem Traffic Counting dengan Algoritma You Only Look Once (YOLO)v4," *J. Media Inform. Budidarma*, vol. 5, no. 4, p. 1438, 2021, doi: 10.30865/mib.v5i4.3309.
- [5] A. A. M. Suradi, M. F. Rasyid, and N. Nasaruddin, "Sistem Perhitungan Jumlah Kendaraan Berbasis Computer Vision," *Pros. Semin. Ilm. Sist. Inf. Dan Teknol. Inf.*, vol. XI, no. 1, pp. 89–97, 2022.
- [6] M. D. Adhiwijaya, "Deteksi Helm Sepeda Motor Berbasis Web dengan Algoritma You Only Look Once (YOLO) dengan microframework Flask," <https://repository.pnj.ac.id/id/eprint/4042>, p. 58, 2021.
- [7] Q. Hidayati, "Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 6, no. 2, 2017, doi: 10.22146/jnteti.v6i2.318.
- [8] H. H. Al Asyhar, S. A. Wibowo, and G. Budiman, "Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video," *eProceedings Eng.*, vol. 7, no. 2, p. 3631, 2020.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [10] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018.
- [11] F. Rachmawati and D. Widhyaestoeti, "Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO," *Pros. LPPM UIKA Bogor*, pp. 360–370, 2020.
- [12] Marzuman, Muharnis, Azizul, D. M. Rinaldi, and B. Prasetyo, "Perbandingan Metode Haar Cascade , YoloV3 , dan TinyYoloV3 Dalam Mendeteksi Kendaraan Bermotor Berbasis Video," *Proceeding Appl. Bus. Eng. Conf.*, no. November, pp. 17–19, 2022.
- [13] D. Alfiyandi Abdurrafi, T. Alawiy, and B. M. Basuki, "Deteksi Klasifikasi Dan Menghitung Kendaraan Berbasis Algoritma You Only Look Once (YOLO) Menggunakan Kamera CCTV," no. 1, pp. 1–6, 2023.
- [14] F. Azimah and K. Rizky Nova Wardani, "Sistem Pendeteksi Gejala Awal Covid-19 dengan Penggunaan Metode AI Project Cycle," *J. Locus Penelit. dan Pengabd.*, vol. 1, no. 6, pp. 405–418, 2022, doi: 10.58344/locus.v1i6.135.
- [15] J. Ofoeda, R. Boateng, and J. Effah, "Application programming interface (API) research: A review of the past to inform the future," *Int. J. Enterp. Inf. Syst.*, vol. 15, no. 3, pp. 76–95, 2019, doi: 10.4018/IJEIS.2019070105.
- [16] A. Faqih, K. Mutmainnah, and M. A. R, "Jurnal Teknik Informatika dan Elektro Separation Deteksi Kendaraan Pada Citra Digital Dengan Menggunakan Algoritma YOLO (You Only Look Once)," *J. Tek. Inform. dan Elektro*, vol. 3, no. 2, pp. 5–11, 2021.
- [17] M. K. Suryadewiansyah and T. E. E. Tju, "Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert," *J. Nas. Teknol. dan Sist. Inf.*, vol. 8, no. 2, pp. 81–88, 2022, doi: 10.25077/teknosi.v8i2.2022.81-88.
- [18] A. H. Pratomo, W. Kaswidjanti, and S. Mu'arifah, "Implementasi Algoritma Region Of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan Implementation of Region Of Interest (ROI) Algorithm To Improve Car Detection And Classification Algorithm," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 155–162, 2020, doi: 10.25126/jtiik.202071718.