JIMU: Jurnal Ilmiah Multi Disiplin
Vol. 03, No. 02, Tahun 2025, Hal. 889 - 910, ISSN: 3031-94198 (Online)

Pengembangan Sistem Deteksi Kecepatan Kendaraan *Overspeed* dengan YOLOv8 di Area Jalan Tol

A. Arfan Fauzi[1], Abdul Wahid[2], Andi Baso Kaswar[3]
[123]*Universitas Negeri Makassar*
Email: arfanfauzi03@gmail.com

Article Info

Article history:

Received January 24, 2025
Revised February 10, 2025
Accepted February 24, 2025

Keywords:

YOLO, Speed Detection, Traffic Monitoring, Computer Vision, Deep Learning

ABSTRACT

Toll roads in big cities like Makassar play an important role in community mobility, but overspeed violations often occur, threatening the safety of road users. This research aims to develop an overspeed detection system using YOLOv8 algorithm with video input from CCTV. A dataset of 300 images obtained from CCTV recordings of the Makassar toll road area which were then converted into images was used for model training. The training results show that the YOLOv8n model with AdamW optimiser, image size 640x640 pixels, number of batches 16, and number of epochs 100 times provides the best performance with mAP50 value of 0.992 and mAP50-95 of 0.897. System test results in detecting objects in the form of vehicles show 100% accuracy in various lighting conditions and camera positions. The system is also able to detect vehicle speed with a small percentage error, the percentage error ranges from 1.67% to 7.5%, and an average percentage error of 3.97%. This system can identify overspeed vehicles accurately and is implemented into a Flask-based website for real-time monitoring.

Article Info

Article history:

Received January 24, 2025
Revised February 10, 2025
Accepted February 24, 2025

Keywords:

YOLO, Deteksi Kecepatan, Pemantauan Lalu Lintas, Computer Vision, Deep Learning

ABSTRACT

Jalan tol di kota besar seperti Makassar memainkan peran penting dalam mobilitas masyarakat, namun sering terjadi pelanggaran *overspeed* yang mengancam keselamatan pengguna jalan. Penelitian ini bertujuan mengembangkan sistem deteksi kecepatan kendaraan *overspeed* menggunakan algoritma YOLOv8 dengan input video dari CCTV. Dataset berupa 300 citra yang diperoleh dari rekaman CCTV area jalan tol Makassar yang kemudian dikonversi menjadi citra digunakan untuk pelatihan model. Hasil pelatihan menunjukkan model YOLOv8n dengan *optimizer* AdamW, *image size* 640x640 piksel, jumlah *batch* 16 dan jumlah *epoch* sebanyak 100 kali memberikan kinerja terbaik dengan nilai mAP50 sebesar 0.992 dan mAP50-95 sebesar 0.897. Hasil pengujian sistem dalam mendeteksi objek berupa kendaraan menunjukkan akurasi 100% pada berbagai kondisi pencahayaan dan posisi kamera. Sistem juga mampu mendeteksi kecepatan kendaraan dengan persentase *error* yang kecil, persentase *error* berkisar antara 1.67% hingga 7.5% dan rata-rata persentase error sebesar 3.97%. Sistem ini mampu mengidentifikasi kendaraan *overspeed* secara akurat dan diimplementasikan ke dalam website berbasis Flask untuk pemantauan secara *real-time*.

Corresponding Author:
Nama penulis: A. Arfan Fauzi
Universitas Negeri Makassar
Email: arfanfauzi03@gmail.com

Pendahuluan

Jalan tol merupakan salah satu infrastruktur yang membantu meningkatkan kelancaran lalu lintas dan mobilitas kendaraan dalam suatu wilayah (Hermanto et al., 2019)(Agustin & Hariyani, 2023). Lalu lintas jalan tol di kota besar seperti Makassar merupakan salah satu komponen penting dalam mobilitas masyarakat. Namun, salah satu masalah yang seringkali muncul adalah tingginya tingkat pelanggaran kendaraan di jalan tol (Luiza, 2023). Pelanggaran kendaraan dapat mengancam keselamatan pengguna jalan serta meningkatkan potensi terjadinya kecelakaan yang serius.

Berdasarkan data laporan tahunan Badan Pengatur Jalan Tol 2022 (Badan Pengatur Jalan Tol, 2022), tingkat kecelakaan di seluruh jalan tol di Indonesia telah mengalami peningkatan yang signifikan. Pada tahun 2020, dilaporkan terjadi sebanyak 3.907 kecelakaan di jalan tol, angka yang meningkat secara signifikan pada tahun 2021 menjadi 3.988 kecelakaan, dan terus meningkat pada tahun 2022 menjadi 4.487 kecelakaan. Tingginya tingkat kecelakaan di jalan tol ini menjadi perhatian utama dalam upaya meningkatkan keselamatan pengguna jalan.

Salah satu pelanggaran yang menjadi faktor meningkatnya tingkat kecelakaan ialah pelanggaran kendaraan *overspeed* (Enggarsasi & Sa'diyah, 2017). *Overspeed* merupakan kondisi kendaraan bergerak melebihi batas kecepatan yang telah ditetapkan. Pemerintah telah mengatur batas kecepatan di mayoritas ruas jalan tol dalam Peraturan Menteri Perhubungan No. PM 111 Tahun 2015 tentang Tata Cara Penetapan Batas Kecepatan. Peraturan tersebut mengatur bahwa saat berkendara di tol dalam kota, batas kecepatan minimalnya adalah 60 km/jam dan maksimal 80 km/jam, sedangkan di tol luar kota, kecepatan minimalnya 60 km/jam dan maksimal 100 km/jam (Perhubungan, 2015). Pelanggaran kendaraan *overspeed* dapat mengancam keselamatan pengguna jalan dan meningkatkan risiko terjadinya kecelakaan serius. Oleh karena itu, pemantauan terhadap lalu lintas di jalan tol sangatlah penting sebagai upaya untuk mencegah terjadinya kecelakaan lalu lintas, baik disengaja maupun tidak disengaja.

Dalam mengatasi masalah tersebut, diperlukan sebuah sistem untuk mengidentifikasi pelanggaran kendaraan *overspeed* secara otomatis di area jalan tol. Solusi untuk mengatasi permasalahan ini adalah dengan memanfaatkan teknologi deteksi objek. Teknologi deteksi objek (*object detection*) merupakan teknologi *computer vision* yang memiliki kemampuan untuk mengidentifikasi dan melacak suatu objek pada suatu gambar maupun video (Rahmad et al., 2020)(Tiyar & Fudholi, 2021).

Berdasarkan penelitian yang telah dilakukan sebelumnya pada tahun 2020, telah dilakukan penelitian menggunakan metode YOLOv2 dalam mendeteksi kendaraan, sementara

algoritma *Simple Online Realtime Tracking* (SORT) digunakan untuk *tracking* kendaraan. Pada penelitian ini diperoleh hasil evaluasi kinerja dengan nilai RMSE (*root mean square error*) sebesar 0,625 m/s dan MAPE (*mean absolute percentage error*) sebesar 20,922 %. Pada penelitian ini metode yang digunakan masih terdapat kendala dalam mendeteksi kendaraan baik dalam keadaan bergerak maupun berhenti, sehingga berdampak saat proses pendeteksian kecepatan (Bell et al., 2020).

Selanjutnya pada tahun 2021, dilakukan penelitian menggunakan metode *Haar Cascade Classifier* dalam mendeteksi kecepatan kendaraan. Hasil Evaluasi kinerja sistem pada penelitian ini dalam mendeteksi kendaraan yaitu diperoleh nilai rata-rata *recall* 0.988, nilai presisi 0.97, dan hasil dari pengujian perhitungan kecepatan diperoleh nilai *Mean Squared Error* (MSE) sebesar 0,6. Penelitian ini hanya terbatas pada kondisi terang dalam mendeteksi kecepatan, sehingga pada kondisi pencahayaan rendah atau malam hari hasil dari deteksi objek dan estimasi kecepatan kendaraan tidak optimal (Zulfikri et al., 2021).

Penelitian terkait lainnya dilakukan pada tahun 2021 mengenai penentuan kecepatan kendaraan menggunakan metode *Frame Difference*, sedangkan proses segmentasi objek menggunakan *MobileNetV2*. Pada penelitian diperoleh hasil akurasi penentuan titik referensi yang hanya mencapai 74,81%, yang mempengaruhi keandalan perhitungan kecepatan kendaraan. Selain itu, akurasi dalam menghitung kecepatan dengan titik referensi kanan sebesar 83,84% dan titik referensi kiri sebesar 79,81% dinilai belum optimal, terutama dalam skenario yang lebih kompleks atau dengan lalu lintas padat (Tadjudin & Rosmala, 2021).

Penelitian terkait lainnya juga telah dilakukan pada tahun 2021, penelitain ini menggunakan metode *Optical Flow* dalam membangun sistem deteksi kecepatan. Diperoleh hasil akurasi pengukuran kecepatan berkisar 82,5% hingga 96,5%. Dari hasil nlai akurasi pengukuran kecepatan tersebut dapat dinilai bahwa metode yang digunakan masih kurang optimal, sehingga masih perlu dilakukan pengembangan lebih lanjut (Carlos et al., 2021).

Pada penelitian terkait selanjutnya yang dilakukan pada tahun 2021 mengenai sistem pengukur kecepatan kendaraan dengan menggunakan algoritma *Image Subtracting*. Penelitian ini menggunakan algoritma *Image Subtracting* dan metode *Gaussian Mixture Model* (GMM) dalam mendeteksi objek kendaraan. Sistem ini mampu melakukan pengukuran kecepatan kendaraan yang melebihi batas kecepatan maksimal yaitu 50 km/jam. Adapun hasil yang diperoleh selama pengambilan data didapatkan beberapa nilai error yang cukup besar yaitu 23,24%, sehingga penelitian ini masih perlu dikembangkan untuk meningkatkan efektivitas penggunaan video dan gambar dengan metode *image subtracting* (Satura et al., 2021).

Penelitian lain terkait pengukuran kecepatan kendaraan juga dilakukan pada tahun 2023 mengenai sistem pengukur kecepatan kendaraan di jalan tol yang menggunakan metode *Background Subtraction*. Penelitian ini menggunakan menggunakan metode *Background Subtraction* dalam mendeteksi objek dengan cara membandingkan setiap piksel dalam *frame* citra dengan model latar belakang atau *background* yang telah ditentukan sebelumnya. Evaluasi kinerja sistem pada penelitian ini memperoleh hasil *precision* 100%, *recall* 49%, dan *akurasi* 53%. Metode *Background Subtraction* pada penelitian ini kurang mendapatkan hasil yang optimal apabila diimplementasikan pada saat kondisi lalu lintas yang padat (Utama, 2023).

Penelitian terkait lainnya yang dilakukan pada tahun 2023 mengenai deteksi kecepatan kendaraan menggunakan algoritma YOLO pada modul kamera. Penelitian ini menggunakan

algoritma YOLOv5 dalam mendeteksi jenis dan mengukur kecepatan kendaraan yang melintas di jalan raya. Pada penelitian ini hanya mampu mendeteksi dua kelas kendaraan, yaitu mobil dan motor, belum dapat mendeteksi kelas truk dan bus. Rata-rata *error* persen dalam mendeteksi jenis kendaraan adalah 0% untuk mobil dan 9,91% untuk motor, sedangkan *error* persen pembacaan kecepatan kendaraan adalah 4,45% untuk motor dan 10,22% untuk mobil (Faizin, 2023).

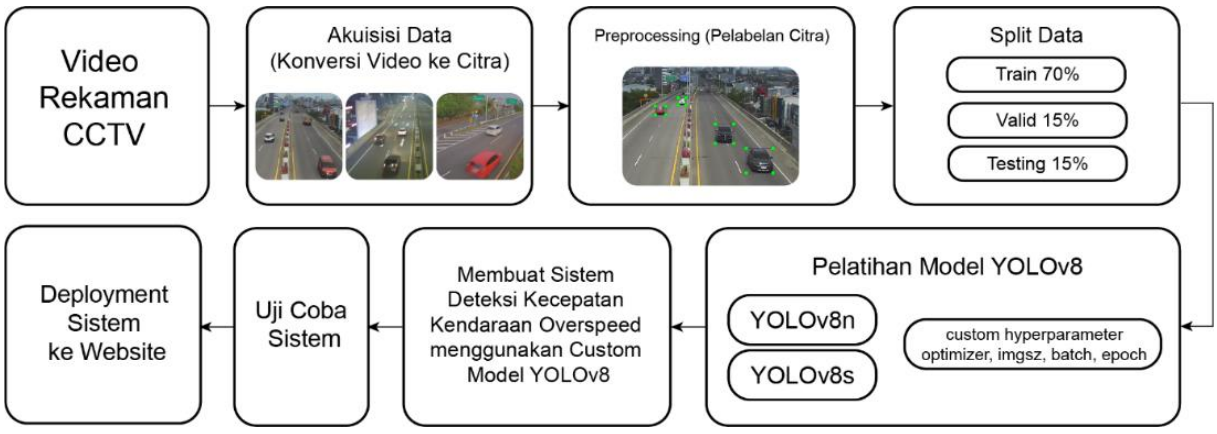
Namun, beberapa penelitian yang telah dijabarkan di atas dalam mendeteksi objek kendaraan dan mendeteksi kecepatan kendaraan, masih terdapat beberapa kelemahan pada metode yang digunakan, baik dalam mendeteksi objek berupa kendaraan maupun untuk mengukur kecepatan kendaraan, sehingga menyebabkan hasil akurasi yang diperoleh kurang optimal.

Oleh karena itu berdasarkan uraian permasalahan diatas, pada penelitian ini diusulkan Sistem Deteksi Kecepatan Kendaraan Overspeed dengan YOLOv8 di Area Jalan Tol. Pada penelitian ini dataset yang digunakan merupakan 300 data citra yang telah dikonversi dari video rekaman CCTV kendaraan melaju di area jalan tol Makassar. Dalam penelitian ini metode yang diusulkan dalam mendeteksi kecepatan kendaraan *overspeed* terdiri dari 7 tahap utama, yaitu: akuisisi citra, *pre-processing*, *split data*, *training* atau pelatihan model deteksi objek berupa kendaraan, pembuatan sistem deteksi kecepatan kendaraan *overspeed*, uji coba sistem, dan *deployment* sistem ke *website* menggunakan *framework* Flask.

Penelitian ini bertujuan untuk mengembangkan teknologi dalam pemantauan lalu lintas dengan memanfaatkan kecerdasan buatan dan pengolahan citra menggunakan algoritma YOLOv8. Dengan demikian, penelitian ini dapat mendeteksi kendaraan *overspeed* di area jalan tol secara *realtime* dan akurat, serta membantu meningkatkan keselamatan lalu lintas dengan memantau dan mengidentifikasi kendaraan *overspeed*. Selain itu, sistem ini diintegrasikan ke dalam website berbasis Flask, sehingga memudahkan proses pemantauan dan evaluasi oleh pihak pengelola jalan tol.

Metode

Metode yang diusulkan dalam penelitian ini terdiri dari beberapa tahap mulai dari akuisisi data sampai dengan deployment sistem ke *website*. Alur metode penelitian sebagaimana pada Gambar 1.



Gambar 1. Alur Metode Pendeteksi Kecepatan yang Diusulkan

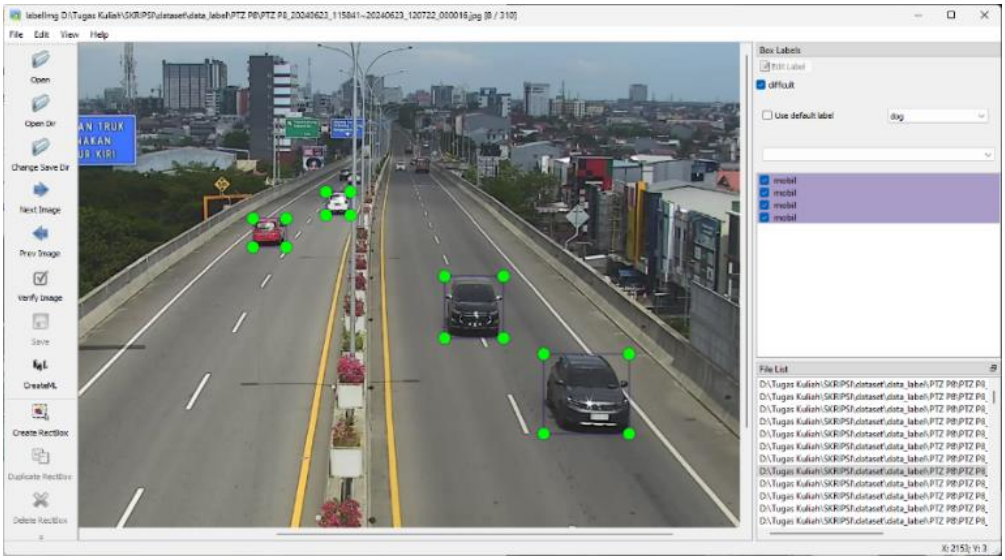
a. Akuisisi Data

Pada penelitian ini dataset yang digunakan merupakan dataset primer yang diperoleh dari *Traffic Information Service* (TIS) berupa video rekaman di beberapa titik CCTV area jalan tol Makassar. Video tersebut kemudian diekstrak menjadi kumpulan citra, sehingga diperoleh 300 citra dari berbagai sudut pandang kamera dan berbagai kondisi pencahayaan.

Gambar 2. Hasil Ekstraksi dari Video Menjadi Citra

b. *Pre-processing*

Setelah dataset berupa 300 citra kendaraan di area jalan tol terkumpul, dilakukan proses *pre-processing* data dengan melabeli citra menggunakan software LabelImg, sebagaima pada Gambar 3. Pelabelan ini bertujuan untuk memberikan kotak pembatas (bounding box) dan keterangan kelas pada objek yang akan dideteksi. Kelas yang digunakan hanya satu, yaitu mobil (termasuk truk dan bus). Proses pelabelan ini penting agar model YOLOv8 dapat mempelajari dan mengenali objek dengan baik selama pelatihan (Maleh et al., 2023)(Guntara, 2023).



Gambar 3. Proses Pelabelan Data

c. *Split Data*

Setelah melalui tahap *pre-processing* pada data, selanjutnya dilakukan pembagian data atau *split data* dengan membagi dataset menjadi tiga bagian, yaitu data latih, data validasi dan data uji. Pembagian data latih, data validasi, dan data uji menggunakan perbandingan 70:15:15, 70% atau 210 data dari dataset digunakan untuk melatih model, 15% atau 45 data dari dataset digunakan untuk proses validasi, dan 15% atau 45 data dari dataset digunakan untuk menguji kinerja dari model yang telah dilatih (Umri & Delica, 2021)(Winanto et al., 2023).

d. Pelatihan Model YOLOv8

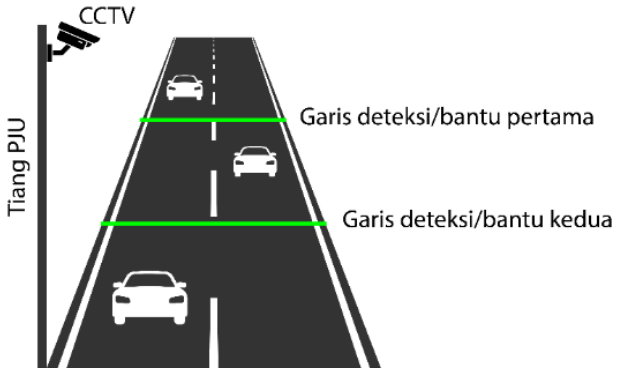
Setelah proses pembagian data, kemudian dilakukan tahap pelatihan model. Tahapan ini bertujuan untuk melatih model YOLOv8. Proses pelatihan ini bertujuan untuk meminimalkan *loss function* dan meningkatkan akurasi prediksi model (Danukusumo, 2017). Pada tahap ini juga dilakukan beberapa skenario pelatihan yang membandingkan *hyperparameter* seperti *Optimizer*, *Image Size*, jumlah *Batch*, dan jumlah *Epoch* yang disesuaikan untuk mengoptimalkan kinerja model (Liao et al., 2022). Model YOLOv8 yang digunakan dalam proses pelatihan yaitu YOLOv8n dan YOLOv8s karena keuda varian model tersebut lebih optimal ketika digunakan secara *realtime* (Sudharson et al., 2023)(Wang et al., 2023)(Ma et al., 2024). Setiap kombinasi *hyperparameter* dengan berbagai variasi ini bertujuan untuk memperoleh nilai metrik mAP yang akan dijadikan sebagai acuan akurasi model YOLO yang akan dibangun, serta *runtime* atau waktu komputasi yang kemudian akan dijadikan perbandingan model yang paling optimal untuk digunakan ke dalam sistem.

Nilai Mean Average Precision (mAP) terbagi menjadi mAP0.5 dan mAP0.5:0.95. Nilai mAP0.5 mengukur rata-rata presisi pada IoU (Intersection over Union) threshold 0.5, menunjukkan kemampuan model mendeteksi objek dengan tepat pada threshold yang lebih rendah. Sementara itu, mAP0.5:0.95 mengukur rata-rata presisi pada berbagai threshold IoU dari 0.5 hingga 0.95, memberikan gambaran lebih komprehensif tentang kinerja model dengan berbagai tingkat *overlap* yang lebih ketat. Kedua metrik ini bersama-sama memberikan evaluasi menyeluruh tentang performa model dalam mendeteksi objek (Ultralytics, 2023)(Yao et al., 2022).

e. Pembuatan Sistem Deteksi Kecepatan Kendaraan *Overspeed* dengan YOLO

Setelah membangun model YOLOv8, kemudian dilakukan pembuatan sistem yang mampu mendeteksi kecepatan kendaraan *overspeed* meggunakkan model YOLO yang telah dilatih dalam pendeteksian kendaraan.

Sebelum ke tahap pembuatan sistem secara menyeluruh, dilakukan perancangan sistem pengambilan data *realtime* untuk menghitung kecepatan kendaraan sebagaimana pada Gambar 4.



Gambar 4. Desain Sistem Pengambilan Data Deteksi Kecepatan

Penempatan kamera IP CCTV pada sistem monitoring deteksi kecepatan kendaraan *overspeed* nantinya seperti pada Gambar 4, di mana kamera CCTV ditempatkan pada ketinggian yang tetap pada tiang PJU (Penerangan Jalan Umum). Dua buah garis deteksi (ROI) digunakan untuk mengestimasi kecepatan kendaraan dengan mengukur waktu yang dibutuhkan objek untuk bergerak dari satu *frame* ke *frame* berikutnya melewati kedua garis deteksi. Adapun

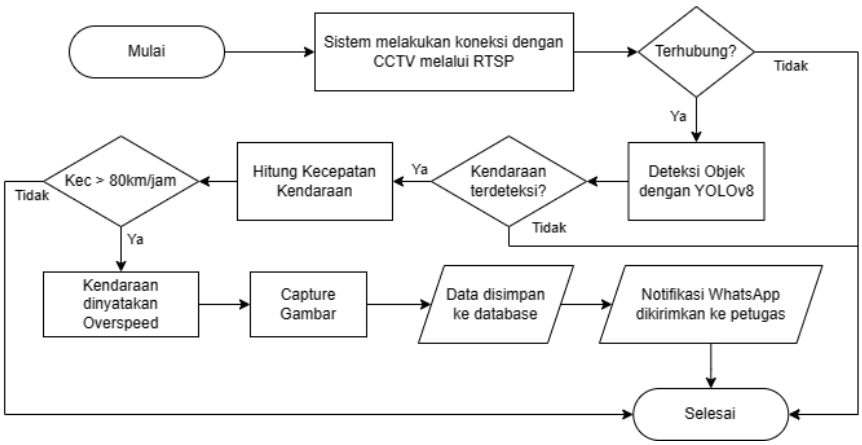
persamaan yang digunakan untuk memperoleh estimasi kecepatan kendaraan, sebagaimana pada persamaan 1 (Satura et al., 2021)(Biassari & Putri, 2021)(Darmawan et al., 2020).

$$\boldsymbol{V = \frac{s}{t}} \quad (1)$$

Keterangan :
V = Kecepatan
s = Jarak sebenarnya antara kedua garis bantu
t = Waktu yang diperlukan objek melewati kedua garis bantu

Nilai jarak diperoleh dari jarak yang sebenarnya dari kedua garis deteksi (ROI). Waktu yang digunakan untuk menghitung estimasi kecepatan kendaraan dihasilkan dari jumlah *frame* yang dihasilkan oleh objek yang terdeteksi saat melewati garis deteksi pertama hingga garis deteksi kedua. Estimasi kecepatan kendaraan akan muncul saat objek yang terdeteksi melintasi kedua garis deteksi (ROI).

Setelah perancangan sistem pengambilan data deteksi kecepatan dilakukan, kemudian dilakukan tahap pembuatan sistem secara menyeluruh. Adapun *flowchart* sistem dapat dilihat pada Gambar 5.



Gambar 5. *Flowchart* Sistem Deteksi Kecepatan Kendaraan *Overspeed*

Proses akan dimulai dengan terlebih dahulu menghubungkan sistem dengan CCTV menggunakan protokol RTSP. Setelah koneksi berhasil, maka proses dilanjutkan pada proses pendeteksian objek berupa kendaraan. Hasil deteksi ini akan diteruskan ke algoritma yang menghitung kecepatan kendaraan. Jika kecepatan kendaraan yang telah terdeteksi melewati 80km/jam, maka kendaraan tersebut akan dinyatakan *overspeed* dan sistem akan melakukan *capture* gambar yang didalamnya terdapat gambar kendaraan dan kecepatan kendaraan tersebut. Setelah itu sistem akan mengirimkan pesan WhatsApp yang berisi waktu, lokasi, jenis pelanggaran dan gambar kendaraan yang telah di-*capture*. Proses ini akan terus berjalan selama sistem tetap terkoneksi dengan CCTV melalui RTSP. Jika koneksi terputus, maka proses akan selesai.

f. Uji Coba Sistem Deteksi Kecepatan Kendaraan Overspeed

Pengujian dilakukan untuk mengetahui akurasi model dalam pendeteksian serta memastikan bahwa metode yang diusulkan dalam membangun sistem mampu mendeteksi kendaraan *overspeed* sebagaimana kondisi sebenarnya.

g. *Deployment* Sistem ke *Website*

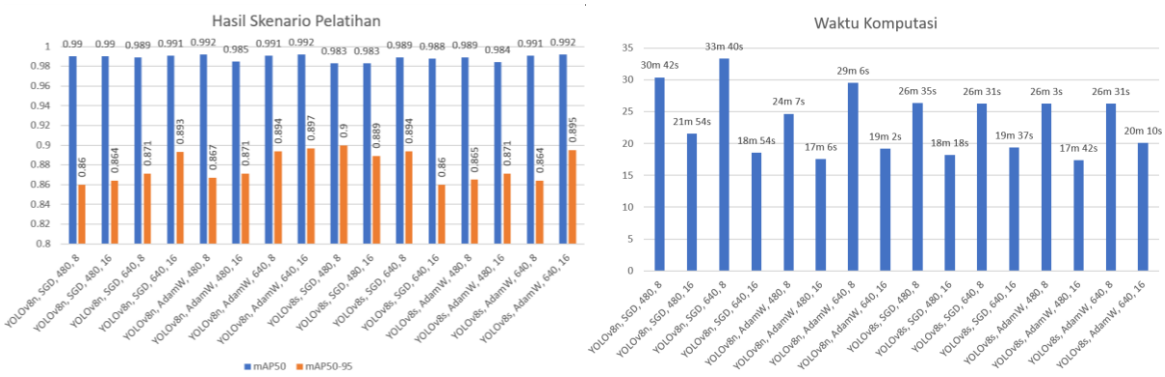
Pada tahap ini, sistem monitoring akan diimplementasikan ke dalam *website* dengan memanfaatkan *framework* Flask (Singh et al., 2019)(Ghimire, 2020), sehingga admin pengelola jalan tol dapat dengan mudah menjalankan dan memantau CCTV. Pada halaman ini, akan ditampilkan antarmuka utama pemantauan CCTV beserta data pelanggaran kendaraan *overspeed* yang terjadi.

Hasil dan Pembahasan

Sebelum membangun sistem, dilakukan beberapa skenario pelatihan untuk membandingkan model YOLOv8 yang akan diimplementasikan ke dalam sistem. Model yang akan dibandingkan ialah YOLOv8n dan YOLOv8s yang akan dikombinasikan dengan berbagai variasi *hyperparameter*. Proses pelatihan model algoritma YOLOv8 yang mampu mendeteksi objek berupa kendaraan dilakukan menggunakan platform berbasis *website*, yaitu Google Colab. Adapun hasil dari beberapa skenario pelatihan yang telah dilakukan sebagaimana pada Tabel 1.

Tabel 1. Hasil Skenario Pelatihan Model

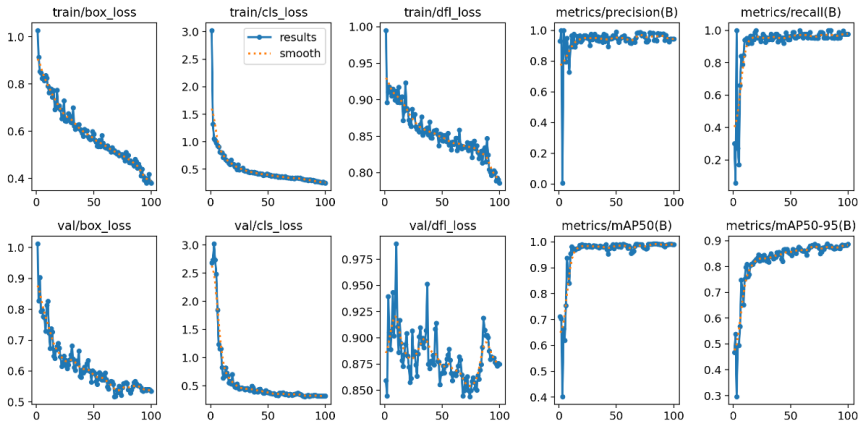
Model	Optimizer	IMG	Batch	Epoch	mAP50	mAP50-95	Waktu Komputasi
YOLOv8n	SGD	480	8	100	0.99	0.86	30m 42s
			16	100	0.99	0.864	21m 54s
		640	8	100	0.989	0.871	33m 40s
			16	100	0.991	0.893	18m 54s
	AdamW	480	8	100	0.992	0.867	24m 7s
			16	100	0.985	0.871	17m 6s
		640	8	100	0.991	0.894	29m 6s
			16	100	0.992	0.897	19m 2s
YOLOv8s	SGD	480	8	100	0.983	0.9	26m 35s
			16	100	0.983	0.889	18m 18s
		640	8	100	0.989	0.894	26m 31s
			16	100	0.988	0.86	19m 37s
	AdamW	480	8	100	0.989	0.865	26m 3s
			16	100	0.984	0.871	17m 42s
		640	8	100	0.991	0.864	26m 31s
			16	100	0.992	0.895	20m 10s



(a) (b)

Gambar 6. Grafik Hasil Pelatihan Model (a) Nilai mAP50 dan mAP50-95 Pelatihan Model (b) Waktu Komputasi Pelatihan Model

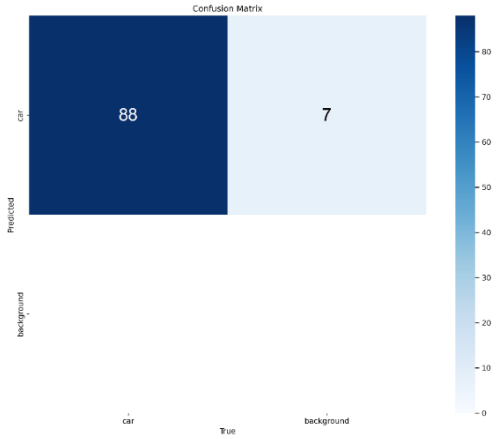
Berdasarkan hasil skenario pelatihan model YOLOv8 dengan berbagai macam variasi kombinasi parameter, penulis memutuskan untuk menggunakan model YOLOv8n dengan konfigurasi parameter *optimizer* AdamW, *image size* 640x640 piksel, jumlah *batch* 16 dan *epoch* sebanyak 100 untuk dijadikan model yang akan digunakan kedalam sistem dalam mendeteksi objek kendaraan. Model YOLOv8n dengan variasi kombinasi parameter tersebut dipilih karena model ini menunjukkan kinerja terbaik dengan nilai mAP50 sebesar 0.992 dan mAP50-95 sebesar 0.897. Selain itu, *runtime* model ini sebesar 19 menit 2 detik cukup efisien yang memungkinkan pelatihan selesai dalam waktu yang wajar tanpa mengorbankan akurasi. Berikut merupakan grafik kinerja dari model YOLOv8n terpilih saat proses training, sebagaimana pada Gambar 7.



Gambar 7. Grafik Kinerja Hasil Pelatihan Model YOLOv8n Terpilih

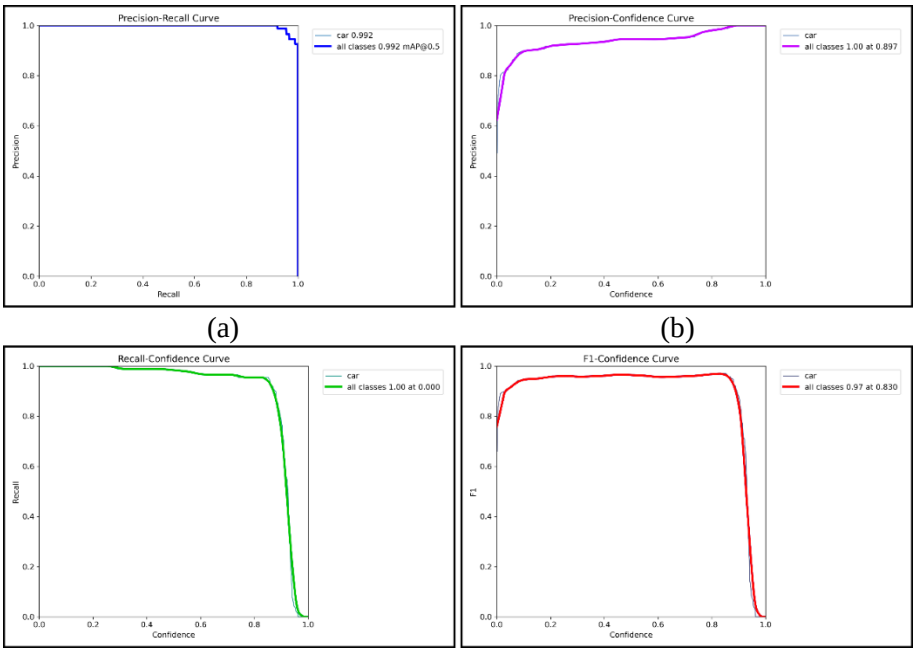
Setelah melakukan proses pelatihan model YOLOv8n terpilih, kemudian diperoleh hasil grafik kinerja model selama proses pelatihan yang menunjukkan *plot box*, *objectness*, *classification*, *precision*, *recall* setiap periode untuk *training* dan validasi dataset. Grafik *train/box_loss* menunjukkan penurunan dari 1.0 menjadi 0.4, menandakan model semakin baik dalam memprediksi *bounding box*. *Train/cls_loss* turun dari 3.0 menjadi 0.5, menunjukkan peningkatan klasifikasi objek. *Train/dfl_loss* menurun dari 1.0 menjadi 0.8, menunjukkan fokus yang lebih baik pada distribusi prediksi. *Metrics/precision(B)* dan *metrics/recall(B*) mendekati 1.0 setelah 10 epochs, menandakan tingkat kesalahan positif yang rendah dan kemampuan

deteksi objek yang baik. Pada data validasi, *val/box_loss* turun dari 1.0 menjadi 0.5, menandakan kinerja baik. *Val/cls_loss* turun dari 3.0 menjadi 0.5, menunjukkan kemampuan generalisasi yang baik. *Val/dfl_loss* sedikit menurun dari 1.0 menjadi 0.85. *Metrics/mAP50(B)* mendekati 1.0, menandakan kinerja sangat baik, sementara *metrics/mAP50-95(B)* mencapai 0.89, menunjukkan konsistensi kinerja pada berbagai threshold. Secara keseluruhan, grafik-grafik ini menunjukkan penurunan konsisten pada berbagai jenis *loss* serta peningkatan signifikan pada metrik presisi, *recall*, dan *mean Average Precision*, baik pada data pelatihan maupun data validasi.



Gambar 8. *Confussion Matrix*

Confusion matrix hasil proses pelatihan model pada Gambar 8, menunjukkan bahwa model mampu mengidentifikasi mobil dengan 88 prediksi benar (*True Positives*) dari 95 total prediksi (88 TP + 7 FP), meskipun ada beberapa kesalahan dalam mengidentifikasi background sebagai mobil.



(a) (b)

(c) (d)

Gambar 9. Kurva Nilai Metrik pada Model (a) *Precision* terhadap *Recall* (b) *Precision* terhadap *Confidence* (c) *Recall* terhadap *Confidence* (d) *F1* terhadap *Confidence*

Gambar 9(a) merupakan grafik *precision* terhadap *recall* yang dihasilkan saat proses validasi model. Garis biru tebal menunjukkan bahwa *mean Average Precision* (mAP) pada *threshold* 0.5 untuk semua kelas adalah 0.992. Model YOLOv8 menunjukkan kinerja yang sangat baik dalam mendeteksi objek kendaraan. Hal ini dibuktikan dengan nilai *mean Average Precision* (mAP) sebesar 0.992 pada threshold 0.5. Kurva *precision-recall* yang hampir mendekati sudut kanan atas grafik menunjukkan bahwa model memiliki tingkat *precision* dan *recall* yang tinggi, menandakan kemampuan model dalam mengidentifikasi *true positives* dengan akurasi yang tinggi serta meminimalkan jumlah *false positives*. Selain itu, nilai mAP yang tinggi juga mengindikasikan bahwa model mampu mempertahankan konsistensi kinerjanya. Dengan kemampuan ini, model YOLOv8 terbukti memiliki performa yang sangat baik dan andal untuk diterapkan pada aplikasi deteksi objek dalam berbagai kondisi pencahayaan dan situasi, menjadikannya pilihan yang sesuai untuk sistem yang memerlukan deteksi objek secara akurat dan efisien.

Gambar 9(b) merupakan grafik *precision* terhadap *confidence*. Hasil proses pengujian model menunjukkan bahwa nilai presisi mencapai 1 pada tingkat *confidence* 0.897, menunjukkan tingkat kepercayaan yang tinggi dalam prediksi model. Dari grafik tersebut, terlihat bahwa *precision* model meningkat seiring dengan naiknya nilai *confidence*, dan pada *confidence* sekitar 0.897, *precision* mencapai nilai maksimum 1.0 untuk semua kelas. Garis kurva yang cenderung mendatar setelah nilai *confidence* mencapai sekitar 0.6 hingga 0.9 menunjukkan bahwa pada level *confidence* tersebut, model dapat dengan sangat andal mengklasifikasikan objek tanpa menghasilkan banyak kesalahan (false positives). Ini menunjukkan bahwa model memiliki tingkat kepercayaan yang tinggi dalam membuat keputusan klasifikasi pada tingkat *confidence* tertentu, yang sangat penting dalam aplikasi yang memerlukan deteksi objek dengan tingkat akurasi yang tinggi. Kemampuan model untuk mencapai *precision* 1.0 pada *confidence* 0.897 juga mengindikasikan bahwa pada titik ini, model hampir tidak membuat kesalahan dalam deteksi, menjadikannya sangat efektif dan andal dalam tugas deteksi objek yang dihadapi. Hal ini menunjukkan keandalan model YOLOv8 dalam menghasilkan prediksi yang akurat pada berbagai tingkat *confidence*, yang sangat berguna ketika diimplementasikan kedalam sistem, di mana keputusan deteksi yang tepat sangat penting.

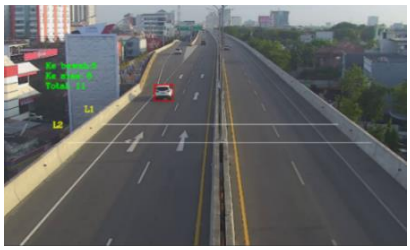
Gambar 9(c) merupakan grafik *recall* terhadap *confidence*. Hasil proses pengujian model menunjukkan *recall* mencapai 1 pada tingkat *confidence* 0.000, mengindikasikan kemampuan model dalam mendeteksi semua instance positif pada berbagai tingkat kepercayaan. Dari grafik tersebut, terlihat bahwa pada tingkat *confidence* 0.000, nilai *recall* mencapai angka maksimal, yaitu 1 untuk semua kelas. Ini mengindikasikan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi hampir semua objek pada tingkat kepercayaan yang rendah, memastikan bahwa sedikit atau bahkan tidak ada objek yang terlewat. Ketika nilai *confidence* meningkat, kurva menunjukkan sedikit penurunan, namun tetap stabil hingga sekitar nilai *confidence* 0.8. Penurunan yang terjadi setelah titik ini menunjukkan peningkatan kepercayaan model dalam deteksinya, di mana model menjadi lebih selektif dalam memilih objek yang dianggap valid. Penurunan pada *recall* seiring dengan peningkatan *confidence*, terutama setelah melewati ambang 0.8, menggambarkan adanya keseimbangan yang dijaga oleh model antara

recall dan *precision*. Meskipun *recall* sedikit berkurang, ini tidak serta-merta menunjukkan performa yang buruk, melainkan bahwa model mulai fokus pada deteksi yang lebih akurat dan mengurangi kemungkinan kesalahan deteksi. Dengan kata lain, model lebih berhati-hati dan lebih percaya diri dalam mendeteksi objek berupa kendaraan, terutama ketika diimplementasikan kedalam sistem, merupakan karakteristik yang diinginkan untuk meningkatkan keandalan sistem secara keseluruhan. Kurva ini menunjukkan bahwa model berhasil menjaga kinerja yang baik dalam mendeteksi objek, dengan hanya sedikit penurunan *recall* pada tingkat *confidence* yang lebih tinggi, yang menunjukkan kemampuan adaptasi model terhadap berbagai tingkat kepercayaan dalam pengambilan keputusan.

Gambar 9(d) merupakan grafik *F1-score* yang menunjukkan skor F1 mencapai 0.97 pada nilai *confidence* 0.830, menandakan keseimbangan yang sangat baik antara presisi dan *recall*. Berdasarkan grafik tersebut, terlihat bahwa skor F1 meningkat dengan cepat seiring bertambahnya *confidence* hingga mencapai nilai sekitar 0.830, di mana F1 mencapai puncaknya pada 0.97. Setelah titik ini, kurva mulai mendatar, yang menunjukkan bahwa model mencapai titik keseimbangan optimal antara presisi dan *recall*. Meskipun pada *confidence* yang sangat tinggi, skor F1 sedikit menurun, hal ini masih dalam batas wajar dan tidak mengindikasikan masalah serius. Penurunan ini dapat diartikan sebagai trade-off alami dalam model yang dapat terjadi ketika *confidence* mendekati nilai maksimum. Secara keseluruhan, grafik ini mengindikasikan bahwa model memiliki kinerja yang sangat baik dalam mengklasifikasikan objek pada berbagai tingkat *confidence*, dengan optimal pada nilai *confidence* sekitar 0.830.

Secara keseluruhan, hasil evaluasi kinerja menunjukkan bahwa model yang digunakan sangat andal dan mampu mempertahankan akurasi yang tinggi. Ini menandakan bahwa model telah dilatih dengan baik dan dapat diandalkan untuk diaplikasikan dalam sistem yang dirancang, baik untuk deteksi maupun klasifikasi objek di berbagai kondisi uji.

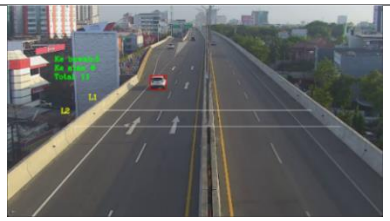
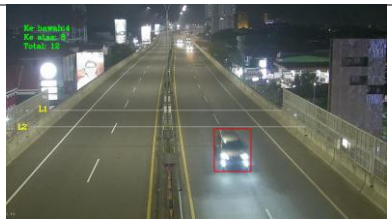
Tabel 2. Hasil Pengujian Pengaruh Posisi Kamera

Posisi Kamera	Hitung Manual	Hasil Deteksi	Akurasi (%)
 Mengarah kedua arah jalan	11	11	100
 Mengarah ke satu arah jalan	9	9	100

Pengujian sistem diawali dengan melakukan pengujian pengaruh posisi kamera terhadap deteksi objek oleh model YOLOv8, hasil yang diperoleh sebagaimana pada Tabel 2. Pada skenario pertama, dengan posisi kamera CCTV mengarah kedua arah jalan, hasil hitung manual dan deteksi sistem sama-sama sebanyak 11 kendaraan (6 ke atas dan 5 ke bawah). Pada skenario kedua, dengan posisi kamera CCTV mengarah ke satu arah jalan, hasil hitung manual dan deteksi sistem juga sama-sama sebanyak 9 kendaraan.

Hasil deteksi kendaraan terhadap posisi kamera dari kedua sudut pandang, keduanya memperoleh akurasi sebesar 100% dalam mendeteksi kendaraan, sehingga dapat dikatakan model yang telah dilatih mampu digunakan dalam sistem yang akan dibangun dalam mendeteksi kendaraan dari kedua sudut pandang, baik itu sudut pandang kamera CCTV mengarah kedua arah jalan, maupun ke satu arah jalan. Hasil tersebut diperoleh karena beberapa faktor. Pertama, data yang digunakan dalam pelatihan model telah mencakup citra dengan berbagai sudut pandang kamera, sehingga model telah dilatih untuk mengenali kendaraan dalam berbagai sudut pandang kamera yang berbeda. Kedua, arsitektur YOLOv8 yang canggih dilengkapi dengan mekanisme pemrosesan gambar yang efektif, memungkinkan model untuk menangkap fitur penting dari kendaraan meskipun terdapat variasi sudut pandang kamera.

Tabel 3. Hasil Pengujian Pengaruh Pencahayaan

Kondisi Pencahayaan	Hitung Manual	Hasil Deteksi	Akurasi (%)
 Pagi	11	11	100
 Siang	19	19	100
 Sore	11	11	100
 Malam	12	12	100

Pada hasil pengujian sistem untuk mengetahui pengaruh kondisi pencahayaan terhadap deteksi objek oleh model YOLOv8, hasil yang diperoleh sebagaimana pada Tabel 3. Pada skenario pagi hari, deteksi model dan hitung manual sama-sama mendeteksi 11 kendaraan (6 ke atas dan 5 ke bawah). Pada skenario siang hari, baik deteksi model maupun hitung manual mendeteksi 19 kendaraan (7 ke atas dan 12 ke bawah). Pada skenario sore hari, deteksi model dan hitung manual sama-sama mendeteksi 11 kendaraan. Pada skenario malam hari, deteksi model dan hitung manual sama-sama mendeteksi 12 kendaraan (8 ke atas dan 4 ke bawah).

Hasil deteksi kendaraan terhadap berbagai kondisi pencahayaan masing-masing memperoleh akurasi sebesar 100% dalam mendeteksi kendaraan, sehingga dapat dikatakan model yang telah dilatih mampu digunakan dalam sistem yang akan dibangun dalam mendeteksi kendaraan pada berbagai kondisi pencahayaan (pagi, siang, sore dan malam). Model yang telah dilatih mampu mempertahankan akurasinya dalam mendeteksi kendaraan pada berbagai kondisi pencahayaan berkat beberapa faktor utama. Pertama, data yang digunakan dalam pelatihan model telah mencakup citra dengan berbagai kondisi pencahayaan, sehingga model telah dilatih untuk mengenali kendaraan dalam berbagai situasi pencahayaan yang berbeda. Kedua, arsitektur YOLOv8 yang canggih dilengkapi dengan mekanisme pemrosesan gambar yang efektif, memungkinkan model untuk menangkap fitur penting dari kendaraan meskipun terdapat variasi pencahayaan.

Pada hasil pengujian sistem dalam mendeteksi kecepatan kendaraan, hasil yang diperoleh sebagaimana pada Tabel 4. Pada pengujian pertama, kecepatan sebenarnya 40 Km/Jam terdeteksi sebagai 37 Km/Jam dengan error 7.5%. Pengujian kedua mendeteksi 59 Km/Jam dari kecepatan sebenarnya 60 Km/Jam dengan error 1.67%. Pada pengujian ketiga, kecepatan 80 Km/Jam terdeteksi sebagai 78 Km/Jam dengan error 2.5%. Terakhir, pengujian keempat menunjukkan deteksi 96 Km/Jam dari kecepatan sebenarnya 100 Km/Jam dengan error 4%.

Secara keseluruhan, hasil pengujian sistem dalam mendeteksi kecepatan kendaraan menunjukkan bahwa sistem mampu mendeteksi kecepatan kendaraan dengan persentase error yang relatif kecil pada berbagai kecepatan kendaraan. Dari keempat pengujian tersebut diperoleh rata-rata persentase *error* sebesar 3.97%. Sistem ini terbukti mampu mendeteksi kecepatan kendaraan dengan cukup baik, meskipun terdapat sedikit perbedaan antara hasil deteksi sistem dan kecepatan sebenarnya. Perbedaan tersebut terjadi karena ketidakakuratan dalam kalibrasi sistem, seperti kesalahan pada penempatan garis deteksi, penentuan jarak ataupun penempatan kamera kurang tepat yang menyebabkan perbedaan antara kecepatan yang terdeteksi oleh sistem dan kecepatan sebenarnya. Oleh karena itu, masih diperlukan perbaikan dalam kalibrasi sistem untuk memperoleh hasil yang lebih akurat dalam mendeteksi kecepatan kendaraan.

Selanjutnya pengujian deteksi kecepatan kendaraan dilakukan secara *realtime* dengan menggunakan kendaraan PJR (Patroli Jalan Raya) jalan tol Makassar dengan kecepatan tertentu yang telah ditentukan. Terdapat 4 skenario pengujian deteksi kecepatan kendaraan, dengan masing-masing kecepatan 40km/jam, 60km/jam, 80km/jam dan 100km/jam. Adapun hasil dari pengujian deteksi kecepatan kendaraan sebagaimana pada Tabel 4.

Tabel 4. Hasil Pengujian Deteksi Kecepatan Kendaraan

Pengujian Ke-	Gambar	Kecepatan Kendaraan Sebenarnya (Km/Jam)	Hasil Deteksi Sistem (Km/Jam)	Error Persen (%)
1		40	37	7,5
2		60	59	1,67
3		80	78	2,5
4		100	96	4
		Rata-Rata Persentase *Error* Deteksi Kecepatan		3,92

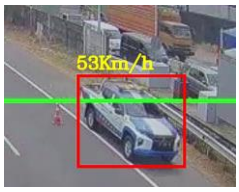
Pada hasil pengujian sistem dalam mendeteksi kecepatan kendaraan, hasil yang diperoleh sebagaimana pada Tabel 4. Pada pengujian pertama, kecepatan sebenarnya 40 Km/Jam terdeteksi sebagai 37 Km/Jam dengan error 7.5%. Pengujian kedua mendeteksi 59 Km/Jam dari kecepatan sebenarnya 60 Km/Jam dengan error 1.67%. Pada pengujian ketiga, kecepatan 80 Km/Jam terdeteksi sebagai 78 Km/Jam dengan error 2.5%. Terakhir, pengujian keempat menunjukkan deteksi 96 Km/Jam dari kecepatan sebenarnya 100 Km/Jam dengan error 4%.

Secara keseluruhan, hasil pengujian sistem dalam mendeteksi kecepatan kendaraan menunjukkan bahwa sistem mampu mendeteksi kecepatan kendaraan dengan persentase error yang relatif kecil pada berbagai kecepatan kendaraan. Dari keempat pengujian tersebut diperoleh rata-rata persentase *error* sebesar 3.97%. Sistem ini terbukti mampu mendeteksi kecepatan kendaraan dengan cukup baik, meskipun terdapat sedikit perbedaan antara hasil deteksi sistem dan kecepatan sebenarnya. Perbedaan tersebut terjadi karena ketidakakuratan dalam kalibrasi sistem, seperti kesalahan pada penempatan garis deteksi, penentuan jarak ataupun penempatan kamera kurang tepat yang menyebabkan perbedaan antara kecepatan yang terdeteksi oleh sistem dan kecepatan sebenarnya. Oleh karena itu, masih diperlukan perbaikan

dalam kalibrasi sistem untuk memperoleh hasil yang lebih akurat dalam mendeteksi kecepatan kendaraan.

Selanjutnya pengujian deteksi kendaraan *overspeed* dilakukan secara *realtime* dengan menggunakan kendaraan PJR (Patroli Jalan Raya) jalan tol Makassar dengan kecepatan tertentu. Mobil PJR yang melaju akan dinyatakan *overspeed* jika melewati batas kecepatan 80Km/jam, sedangkan jika tidak dinyatakan *overspeed*, sistem hanya menampilkan kecepatan mobil PJR yang sebenarnya. Adapun hasil dari pengujian deteksi kendaraan *overspeed* sebagaimana pada Tabel 5.

Tabel 5. Hasil Pengujian Kendaraan Overspeed

Pengujian Ke-	Gambar	Kecepatan Kendaraan (Km/Jam)	Kondisi Sebenarnya	Hasil Deteksi Deteksi Sistem
1		53	Tidak *Overspeed*	Tidak *Overspeed*
2		65	Tidak *Overspeed*	Tidak *Overspeed*
3		85	*Overspeed*	*Overspeed*
4		110	*Overspeed*	*Overspeed*

Setelah melakukan pengujian deteksi kecepatan, kemudian dilakukan pengujian sistem dalam mendeteksi kendaraan *overspeed*, hasil yang diperoleh sebagaimana pada Tabel 5. Pada pengujian pertama, mobil PJR melaju dengan kecepatan 53 km/jam. Sistem dengan benar mendeteksi bahwa mobil PJR tidak *overspeed*, sesuai dengan kondisi sebenarnya. Pengujian kedua mobil PJR melaju dengan kecepatan 65 km/jam. Sama seperti pengujian pertama, sistem berhasil mengidentifikasi bahwa mobil PJR yang melaju tidak *overspeed*. Pada pengujian ketiga, mobil PJR melaju dengan kecepatan lebih tinggi, yaitu 85 km/jam. Sistem mampu mendeteksi bahwa 62 mobil PJR yang melaju dinyatakan *overspeed*, sebagaimana dengan kondisi sebenarnya. Pada pengujian terakhir, mobil PJR melaju dengan kecepatan 110 km/jam. Sistem juga berhasil mendeteksi bahwa mobil PJR yang melaju dinyatakan *overspeed*. Dari hasil pengujian tersebut, dapat disimpulkan bahwa sistem yang dikembangkan mampu

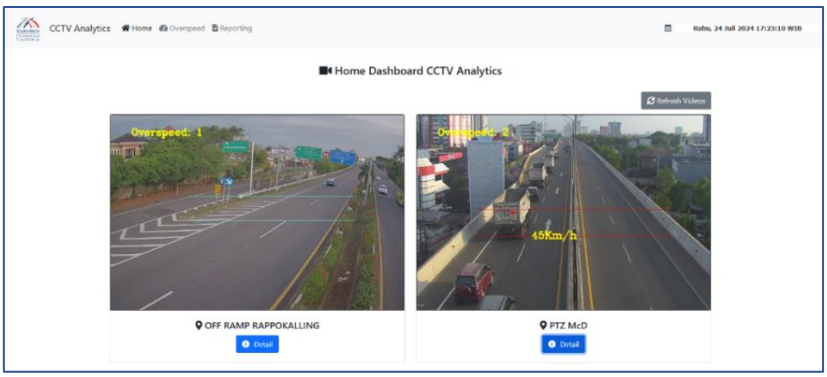
mengidentifikasi kendaraan *overspeed* pada berbagai rentang kecepatan dan memberikan hasil yang sesuai dengan kondisi sebenarnya.

Walaupun hasil tersebut memberikan hasil yang sesuai dengan kondisi sebenarnya, masih terdapat tantangan dalam mendeteksi kendaraan *overspeed.* Salah satu tantangan utama adalah memastikan akurasi deteksi pada berbagai rentang kecepatan. Sistem harus mampu membedakan antara kendaraan yang melaju dalam batas kecepatan yang diizinkan dan yang melampaui batas tersebut. Pada kecepatan rendah, seperti 53 km/jam dan 65 km/jam, deteksi *overspeed* mungkin tidak menampilkan masalah signifikan karena kecepatan kendaraan masih dalam batas yang dapat dengan mudah dikenali. Namun, pada kecepatan yang lebih tinggi, seperti kecepatan diatas 110 km/jam, deteksi overspeed menjadi lebih kompleks karena perbedaan kecil dalam kecepatan mungkin memerlukan tingkat presisi yang lebih tinggi.

Selain itu, faktor-faktor eksternal seperti kondisi pencahayaan, sudut pandang kamera, dan kondisi jalan juga dapat mempengaruhi kemampuan sistem dalam mendeteksi kecepatan kendaraan secara akurat. Misalnya, pada pencahayaan yang buruk atau pada sudut pandang yang tidak ideal, sistem mungkin menghadapi kesulitan dalam menghasilkan pengukuran kecepatan yang konsisten. Oleh karena itu, meskipun sistem ini berhasil mendeteksi kendaraan *overspeed* dengan baik pada rentang kecepatan yang diuji, tetapi pengembangan lebih lanjut masih diperlukan untuk menangani tantangan-tantangan ini, sehingga menghasilkan sistem yang lebih akurat dalam mendeteksi kendaraan *overspeed*.

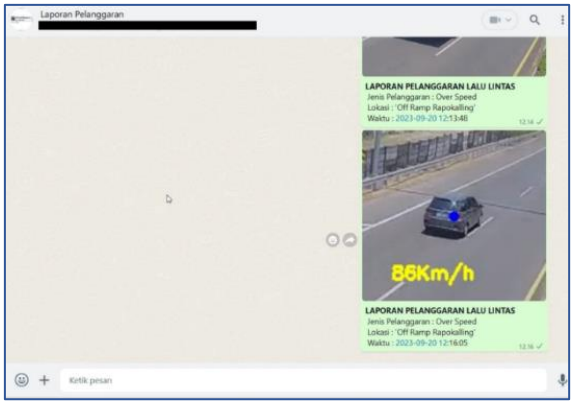
Berdasarkan hasil dari berbagai skenario pengujian sistem, dapat disimpulkan bahwa sistem yang telah dirancang mampu mendeteksi kendaraan yang melewati batas kecepatan (*overspeed*).

Selanjutnya dilakukan proses *deployment* sistem monitoring ke dalam *website*, sehingga admin dapat dengan mudah menjalankan dan memantau CCTV. Pada *website* ini, akan ditampilkan antarmuka utama pemantauan CCTV beserta data pelanggaran kendaraan *overspeed* yang terjadi, sebagaimana pada Gambar 10. Terdapat tiga halaman pada *website* ini, halaman utama, halaman data *overspeed*, dan halaman *reporting*. Halaman utama *website* menampilkan visual monitoring dari beberapa lokasi CCTV dengan sistem deteksi kendaraan overspeed yang menggunakan model YOLOv8 dan dibangun dengan *framework* Flask, sehingga memberikan akses mudah bagi admin untuk memantau aktivitas secara *realtime*. Halaman data *overspeed* menunjukkan rekapitulasi pelanggaran harian, mingguan, dan bulanan, serta grafik pelanggaran berdasarkan lokasi, dan tabel *overspeed* terbaru dengan detail pelanggaran. Terakhir, halaman reporting menyediakan fitur ekspor data dan cetak laporan, sehingga memastikan bahwa seluruh data pelanggaran *overspeed* dapat didokumentasikan dan dianalisis dengan efisien.



Gambar 10. Tampilan Utama Website

Selanjutnya dilakukan penambahan fitur untuk mempermudah monitoring pelanggaran kendaraan *overspeed*. Fitur yang ditambahkan ialah fitur notifikasi pelanggaran *realtime* ke WhatsApp, fitur ini memudahkan dan mempercepat proses pengiriman informasi pelanggaran kendaraan *overspeed* ke petugas divisi lalu lintas yang sedang bertugas yang dapat dilihat pada Gambar 11.



Gambar 11. Notifikasi Pelanggaran Melalui WhatsApp

Pesan yang dikirimkan berisi detail informasi pelanggaran *overspeed* yang terjadi meliputi *capture* kendaraan dan kecepatan pelanggaran yang terjadi, jenis pelanggaran lalu lintas, lokasi dan waktu pelanggaran terjadi.

Terakhir pengujian fungsionalitas dilakukan melalui *black box testing*, yang memusatkan perhatian pada fungsionalitas sistem yang dibangun, terutama pada *input* dan *output* sistem. Pengujian dilakukan dengan beberapa skenario untuk mengetahui fungsionalitas sistem sesuai dengan kebutuhan. Adapun hasil yang diperoleh dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pengujian Black Box Testing

Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan	Hasil Pengujian
Koneksi ke CCTV	Mengkoneksikan CCTV dengan sistem melalui protokol RTSP	Sistem dapat melakukan *streaming* CCTV protokol melalui RTSP	Berhasil

Deteksi Objek Mobil	CCTV menampilkan objek mobil	Sistem dapat mendeteksi objek mobil	Berhasil
Identifikasi Pelanggaran Kendaraan *Overspeed*	CCTV menampilkan kejadian pelanggaran kendaraan *overspeed*	Sistem dapat mengidentifikasi pelanggaran kendaraan *overspeed*	Berhasil
Menyimpan Data ke *Database*	CCTV menampilkan kejadian pelanggaran *overspeed*	Sistem dapat menyimpan data pelanggaran *overspeed* ke *database*	Berhasil
Mengirimkan Notifikasi ke WhatsApp	CCTV menampilkan kejadian pelanggaran kendaraan *overspeed*	Sistem dapat mengirimkan notifikasi *realtime* ke Whatsapp	Berhasil

Hasil pengujian *black box* pada sistem yang dikembangkan menunjukkan bahwa semua skenario pengujian berhasil dilakukan dengan memperoleh hasil sesuai dengan yang diharapkan. Pada skenario koneksi ke CCTV, sistem mampu melakukan streaming CCTV melalui protokol RTSP. Dalam skenario deteksi objek mobil, sistem berhasil mendeteksi objek mobil yang ditampilkan oleh CCTV. Selanjutnya, pada skenario identifikasi pelanggaran kendaraan overspeed, sistem dapat mengidentifikasi kejadian pelanggaran dengan baik. Sistem juga mampu menyimpan data pelanggaran *overspeed* yang terdeteksi ke dalam *database* sesuai dengan yang diharapkan. Pada skenario terakhir, sistem dapat mengirimkan notifikasi secara *realtime* ke WhatsApp mengenai kejadian pelanggaran *overspeed* yang berhasil dideteksi sistem. Hasil ini menunjukkan bahwa secara keseluruhan sistem yang dirancang telah memenuhi seluruh kriteria fungsional yang diharapkan dan siap digunakan dalam pemantauan pelanggaran kendaraan *overspeed* secara efektif.

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa hasil pengujian sistem menunjukkan Sistem Deteksi Kecepatan Kendaraan *Overspeed* dengan YOLOv8 di Area Jalan Tol berhasil dibuat menggunakan metode yang telah diusulkan. Metode yang diusulkan yaitu menggunakan model YOLOv8n dengan konfigurasi *optimizer* AdamW, *image size* 640, *batch* 16 dan epoch sebesar 100. Model ini dipilih karena menghasilkan kinerja yang terbaik diantara model YOLOv8 dengan variasi kombinasi parameter lainnya. Hasil evaluasi kinerja model ini memperoleh nilai metrik seperti *precision* sebesar 96.7%, *recall* sebesar 95.5%, mAP50 sebesar 99.2% dan mAP50-95 sebesar 89.7%, yang menunjukkan kinerja yang sangat baik dalam mendeteksi objek kendaraan. Dengan demikian, sistem ini mampu mengidentifikasi pelanggaran kendaraan overspeed dan memberikan hasil yang cukup akurat. Sistem ini juga terbukti baik dalam mengidentifikasi pelanggaran kendaraan *overspeed*. Hal ini dibuktikan dengan pengujian sistem yang mampu mendeteksi kendaraan dari berbagai sudut pandang kamera dan kondisi pencahayaan dengan akurasi 100%. Pengujian deteksi kecepatan menunjukkan persentase *error* yang relatif kecil pada berbagai kecepatan kendaraan, dengan kesalahan terbesar sebesar 7.5% dan rata-rata persentase error sebesar 3.97%. Sistem juga berhasil mengidentifikasi kendaraan *overspeed* pada berbagai rentang kecepatan dan memberikan hasil yang sesuai dengan kondisi sebenarnya. Selain itu, pengujian *black box* menunjukkan bahwa sistem memenuhi seluruh kriteria fungsional yang diharapkan. Secara keseluruhan, sistem yang dirancang terbukti akurat dan dapat diandalkan dalam mendeteksi dan memantau pelanggaran kendaraan *overspeed* di area jalan tol.

Daftar Pustaka

Agustin, I. W., & Hariyani, S. (2023). *Pengelolaan infrastruktur kota dan wilayah*. Universitas Brawijaya Press.

Badan Pengatur Jalan Tol. (2022). *Laporan Tahunan Badan Pengatur Jalan Tol 2022 : Peningkatan Kualitas Jalan Tol Melalui Inovasi Teknologi Operasi dan Pemeliharaan*. https://bpjt.pu.go.id/booklet/buku_tahunan_BPJT_2022/mobile/index.html

Bell, D., Xiao, W., & James, P. (2020). Accurate Vehicle Speed Estimation from Monocular Camera Footage. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *5*(2), 419–426. https://doi.org/10.5194/isprs-annals-V-2-2020-419-2020

Biassari, I., & Putri, K. E. (2021). Penggunaan Media Video Pembelajaran Interaktif Berbasis Aplikasi Nearpod Pada Materi Kecepatan Di Sekolah Dasar. *Prosiding SEMDIKJAR (Seminar Nasional Pendidikan Dan Pembelajaran)*, *4*, 62–74.

Carlos, F., Margatama, L., & Riyanto, I. (2021). Perancangan Sistem Deteksi Kecepatan Kendaraan Dengan Metode Optical Flow. *Jurnal Ticom: Technology of Information and Communication*, *10*(1), 56–63.

Danukusumo, K. P. (2017). *Implementasi deep learning menggunakan convolutional neural network untuk klasifikasi citra candi berbasis GPU*. UAJY.

Darmawan, C. W., Sompie, S. R. U. A., & Kambey, F. D. (2020). Implementasi Internet of Things pada Monitoring Kecepatan Kendaraan Bermotor. *Jurnal Teknik Elektro Dan Komputer*, *9*(2), 91–100.

Enggarsasi, U., & Sa'diyah, N. K. (2017). Kajian terhadap faktor-faktor penyebab kecelakaan lalu lintas dalam upaya perbaikan pencegahan kecelakaan lalu lintas. *Perspektif: Kajian Masalah Hukum dan Pembangunan*, *22*(3), 238–247.

Faizin, M. (2023). Implementasi Algoritma YOLO pada Modul Kamera untuk Deteksi Jenis dan Kecepatan Kendaraan. In *DIGITAL REPOSITORY UNIVERSITAS JEMBER*. Universitas Jember.

Ghimire, D. (2020). *Comparative study on Python web frameworks: Flask and Django* [Metropolia University of Applied Sciences]. https://urn.fi/URN:NBN:fi:amk-2020052513398

Guntara, R. G. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *5*(1), 55–60.

Hermanto, H., Meiyani, E., & Risfaisal, R. (2019). Persepsi Masyarakat Terhadap Pembangunan Infrastruktur Jalan Tol Layang AP Pettarani Di Kota Makassar. *Equilibrium: Jurnal Pendidikan*, *7*(1), 198–205.

Liao, L., Li, H., Shang, W., & Ma, L. (2022). An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *31*(3), 1–40.

Luiza, S. N. (2023). Analisis Kesadaran Hukum Masyarakat Di Jalan Dalam Berlalu Lintas.

Nomos: Jurnal Penelitian Ilmu Hukum, *3*(4), 128–135.

Ma, B., Hua, Z., Wen, Y., Deng, H., Zhao, Y., Pu, L., & Song, H. (2024). Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. *Artificial Intelligence in Agriculture*, *11*, 70–82. https://doi.org/10.1016/j.aiia.2024.02.001

Maleh, I. M. D., Teguh, R., Sahay, A. S., Okta, S., & Pratama, M. P. (2023). Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan. *Jurnal Informatika*, *10*(1).

Perhubungan, K. (2015). Peraturan Menteri Perhubungan Nomor: 111 Tahun 2015 tentang Tata Cara Penetapan Batas Kecepatan. *Departemen Perhubungan Republik Indonesia, Jakarta*.

Rahmad, E. C., Kom, S. M., Rawansyah, D., Pd, M., Rochastu, T. K., Studi, P., Informatika, T., Informasi, J. T., & Malang, P. N. (2020). Object Detection System Sebagai Alat Bantu Mendeteksi Objek Sekitar untuk Penyandang Tunanetra. *Seminar Informatika Aplikatif Polinema (SIAP)* , 81–88.

Satura, F. R., Chandra, A. A., & Adhinata, F. D. (2021). Pengukur Kecepatan Kendaraan Menggunakan Algoritma Image Subtracting. *Journal ICTEE*, *2*(2), 35–40.

Singh, M., Verma, A., Parasher, A., Chauhan, N., & Budhiraja, G. (2019). Implementation of Database Using Python Flask Framework. *International Journal of Engineering and Computer Science*, *8*(12), 24890–24893.

Sudharson, D., Srinithi, J., Akshara, S., Abhirami, K., Sriharshitha, P., & Priyanka, K. (2023). Proactive Headcount and Suspicious Activity Detection using YOLOv8. *Procedia Computer Science*, *230*(2023), 61–69. https://doi.org/10.1016/j.procs.2023.12.061

Tadjudin, R., & Rosmala, D. (2021). Implementasi MOBILENETV2 dan Frame Difference untuk Penentuan Kecepatan Kendaraan. *Jurnal Ilmiah Teknologi Infomasi Terapan*, *7*(3), 193–204. https://doi.org/10.33197/jitter.vol7.iss3.2021.541

Tiyar, R. I., & Fudholi, D. H. (2021). Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek. *Petir*, *14*(2), 258–268. https://doi.org/10.33322/petir.v14i2.1350

Ultralytics. (2023). *YOLO Performance Metrics*. https://docs.ultralytics.com/guides/yolo-performance-metrics/#class-wise-metrics

Umri, B. K., & Delica, V. (2021). Penerapan Transfer Learning pada Convolutional Neural Networks dalam Deteksi Covid-19. *JNANALOKA*, 53–61.

Utama, G. T. P. W. (2023). *Pengukur Kecepatan Kendaraan di Jalan Tol dengan Metode Background Subtraction (Studi Kasus di Jalan Tol Jatingaleh Semarang)* [UNIVERSITAS ISLAM SULTAN AGUNG]. http://repository.unissula.ac.id/id/eprint/31967

Wang, X., Gao, H., Jia, Z., & Li, Z. (2023). BL-YOLOv8: An improved road defect detection model based on YOLOv8. *Sensors*, *23*(20), 8361.

Winanto, T. S., Rozikin, C., & Jamaludin, A. (2023). Analisa Performa Arsitektur Transfer Learning Untuk Mengindentifikasi Penyakit Daun Pada Tanaman Pangan. *Journal of Applied Informatics and Computing*, *7*(1), 74–87.

Yao, H., Liu, Y., Li, X., You, Z., Feng, Y., & Lu, W. (2022). A detection method for pavement cracks combining object detection and attention mechanism. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), 22179–22189.

Zulfikri, M., Abd Latif, K., Hammad, R., Syahrir, M., & Studi, P. (2021). Deteksi dan Estimasi Kecepatan Kendaraan dalam Sistem Pengawasan Lalu Lintas Menggunakan Pengolahan Citra Detection and Estimation of Vehicle Speed in Traffic Control Systems Using Image Processing. *Agustus*, *20*(3), 455–467.