

Rancang Bangun Sistem Monitoring Pelanggaran Lalu Lintas Berbasis YOLOv8 Studi Kasus Kota Bandar Lampung

Asep Trisna Setiawan, Rian Arizamda, Robby Yuli Endra

Program studi Sistem Informasi, Universitas Bandar Lampung

Program studi Informatika, Universitas Bandar Lampung

Corresponding author: asepiot@ubl.ac.id

Abstract: Traffic accidents in Indonesia have increased significantly, with national cases rising from 103,645 in 2021 to 146,854 in 2023, a cumulative increase of 41.7% in two years, while motorcycles accounted for 76.42% of vehicles involved. In Bandar Lampung City, a strategic transportation node connecting Sumatra and Java, traffic violation monitoring still relies heavily on manual supervision with limited coverage. This study aims to design, implement, and functionally and structurally test a machine learning-based traffic monitoring system using YOLOv8 that integrates detection of four violation types-helmet non-compliance, triple riding, seatbelt non-compliance, and overspeeding-within a single system for the Bandar Lampung context. The system, named "Traffic Monitor," was developed using a modular architecture separating vehicle detection from violation detection, supporting both file upload and live streaming inputs, with user-configurable inference parameters. Black-box testing confirmed that the system's core workflows-streaming, detection, visualization, data storage, and dashboard navigation-function according to defined scenarios, while white-box testing on the DetectionModule indicated low-to-moderate cyclomatic complexity with four independent paths identified. The system successfully met six functional requirements at the interface and workflow level. It is important to note that the custom detection model for the four violation classes has not yet completed training; therefore, quantitative model performance evaluation (precision, recall, F1-score, mAP) is not yet available and remains a direction for future work. These findings indicate that a functionally integrated traffic violation monitoring architecture can be successfully developed and structurally verified prior to full model performance evaluation.

Keyword: traffic monitoring, YOLOv8, traffic violation detection, system development, functional testing

Abstrak: Kecelakaan lalu lintas di Indonesia meningkat signifikan, dengan kasus nasional naik dari 103.645 (2021) menjadi 146.854 (2023), kenaikan kumulatif 41,7% dalam dua tahun, sementara sepeda motor terlibat dalam 76,42% kendaraan yang mengalami kecelakaan. Di Kota Bandar Lampung, simpul transportasi strategis penghubung Sumatra-Jawa, pengawasan pelanggaran lalu lintas masih sangat bergantung pada pemantauan manual dengan jangkauan terbatas. Penelitian ini bertujuan merancang, mengimplementasikan, dan menguji secara fungsional dan struktural sebuah sistem monitoring lalu lintas berbasis machine learning menggunakan YOLOv8 yang mengintegrasikan deteksi empat jenis pelanggaran tidak menggunakan helm, bonceng tiga, tidak menggunakan sabuk pengaman, dan kecepatan berlebih dalam satu sistem terpadu untuk konteks Kota Bandar Lampung. Sistem yang dinamai "Traffic Monitor" dikembangkan dengan arsitektur modular yang memisahkan deteksi kendaraan dari deteksi pelanggaran, mendukung input berupa unggahan berkas maupun *live streaming*, dengan parameter inferensi yang dapat dikonfigurasi pengguna. Pengujian black-box menunjukkan bahwa alur kerja utama sistem streaming, deteksi, visualisasi, penyimpanan data, dan navigasi dashboard berfungsi sesuai skenario yang ditetapkan, sementara pengujian white-box pada modul *Detection Module* mengindikasikan kompleksitas siklomatik rendah hingga sedang dengan empat jalur independen teridentifikasi. Sistem berhasil memenuhi enam kebutuhan fungsional pada level antarmuka dan

alur kerja. Perlu dicatat bahwa model deteksi kustom untuk keempat kelas pelanggaran belum selesai dilatih, sehingga evaluasi performa model secara kuantitatif (precision, recall, F1-score, mAP) belum tersedia dan menjadi arah penelitian selanjutnya. Temuan ini menunjukkan bahwa arsitektur sistem monitoring pelanggaran lalu lintas yang terintegrasi secara fungsional dapat dikembangkan dan diverifikasi secara struktural sebelum evaluasi performa model dilakukan secara lengkap.

Kata Kunci: monitoring lalu lintas, YOLOv8, deteksi pelanggaran lalu lintas, pengembangan sistem, pengujian fungsional

Pendahuluan

Transportasi jalan raya merupakan tulang punggung mobilitas masyarakat perkotaan di Indonesia, namun peningkatan volume kendaraan yang tidak diimbangi dengan kepatuhan berlalu lintas telah menimbulkan risiko keselamatan yang serius. Data Korlantas Polri (2024) mencatat kenaikan jumlah kasus kecelakaan lalu lintas nasional dari 103.645 kasus pada tahun 2021 menjadi 139.258 kasus pada tahun 2022, dan meningkat lagi menjadi 146.854 kasus pada tahun 2023 kenaikan kumulatif sebesar 41,7% dalam dua tahun [1]. Pada tahun 2023 saja, kecelakaan tersebut menyebabkan 18.357 korban meninggal dunia, 11.689 korban luka berat, dan 134.811 korban luka ringan, dengan kerugian materiil diperkirakan mencapai Rp500 miliar. Sepeda motor tercatat terlibat dalam 76,42% dari total kendaraan yang mengalami kecelakaan, menjadikannya kontributor utama terhadap statistik tersebut [1].

Kota Bandar Lampung, sebagai simpul transportasi strategis yang menghubungkan jalur Sumatra-Jawa melalui Pelabuhan Bakauheni, menghadapi volume lalu lintas yang tinggi namun pengawasan kepatuhan berlalu lintas masih sangat bergantung pada pemantauan manual oleh petugas di lapangan. Keterbatasan sumber daya manusia dan jangkauan pengawasan menyebabkan banyak pelanggaran seperti tidak menggunakan helm, membawa penumpang berlebih (bonceng tiga), tidak menggunakan sabuk pengaman, dan melaju dengan kecepatan berlebih tidak dapat terdeteksi dan ditindak secara konsisten.

Perkembangan deteksi objek berbasis deep learning telah mengalami evolusi pesat sejak diperkenalkannya arsitektur region-based seperti Faster R-CNN [2] dan pendekatan single-shot seperti SSD [3], hingga munculnya keluarga YOLO (You Only Look Once) yang menawarkan deteksi objek secara real-time dengan satu tahap inferensi tunggal [4]. Arsitektur YOLO terus disempurnakan melalui versi-versi berikutnya, termasuk YOLOv4 yang meningkatkan keseimbangan antara akurasi dan kecepatan [5], hingga YOLOv8 yang membawa peningkatan signifikan pada desain backbone, neck, dan head secara anchor-free [6]. Kematangan arsitektur YOLOv8 ini membuka peluang otomatisasi deteksi pelanggaran lalu lintas secara real-time pada konteks pengawasan jalan raya.

Beberapa penelitian terdahulu telah menunjukkan potensi ini pada domain yang lebih sempit. Neamah dan Karim [7] mengembangkan sistem deteksi dan tracking kendaraan berbasis YOLOv8 dengan akurasi deteksi dan tracking 96,58%, akurasi vehicle counting 97,54%, akurasi estimasi ukuran 96,58%, dan akurasi estimasi kecepatan 87,28%. Shoman et al. [8] mengintegrasikan YOLOv8 dengan deep convolutional generative adversarial network (DCGAN) untuk deteksi pelanggaran penggunaan helm. Sutikno et al. [9] merancang tiga subsistem untuk deteksi kepatuhan sabuk pengaman dan melaporkan bahwa YOLOv8 mengungguli YOLOv5 pada tugas tersebut. Shaqib et al. [10] mengembangkan deteksi kecepatan kendaraan berbasis YOLOv8 dengan pendekatan supervised learning, mencapai mean absolute error 3,5 dan root mean square error 4,22 dibandingkan pengukuran kecepatan aktual. Dou et al. [11] mengusulkan modifikasi YOLOv8 dengan multi-scale feature fusion dan optimized non-maximum suppression yang diuji pada dataset VisDrone2019, sementara Dharmasaputra dan Hartato [12]

mengevaluasi kombinasi YOLOv8 dan centroid tracking untuk deteksi, klasifikasi, dan penghitungan kendaraan. Premaratne et al. [13] menyajikan kajian komprehensif mengenai deteksi, klasifikasi, dan penghitungan kendaraan di jalan raya.

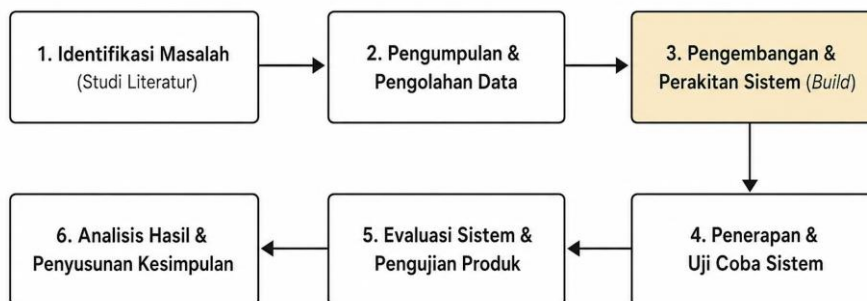
Meninjau penelitian-penelitian tersebut, tampak bahwa mayoritas studi terdahulu memfokuskan sistem pada satu jenis pelanggaran atau satu aspek pemantauan lalu lintas secara terpisah. Se jauh penelusuran yang dilakukan, belum ditemukan penelitian yang mengintegrasikan deteksi empat jenis pelanggaran (helm, bonceng tiga, sabuk pengaman, dan kecepatan berlebih) sekaligus dalam satu sistem monitoring terpadu berbasis YOLOv8 yang dirancang untuk konteks operasional lalu lintas perkotaan di Indonesia. Kesenjangan ini menjadi dasar bagi pengembangan sistem yang diusulkan pada penelitian ini.

Penelitian ini bertujuan untuk merancang, mengimplementasikan, dan menguji secara fungsional dan struktural sebuah sistem monitoring lalu lintas berbasis machine learning menggunakan YOLOv8 yang mendukung deteksi empat jenis pelanggaran lalu lintas secara terintegrasi di Kota Bandar Lampung. Sebagai kontribusi, penelitian ini menghasilkan arsitektur sistem yang modular memisahkan model deteksi kendaraan dan model deteksi pelanggaran serta menerapkan prinsip transparansi status model kepada operator, sehingga sistem dapat digunakan secara bertanggung jawab sekalipun sebagian modul deteksinya masih dalam tahap pengembangan lanjutan. Perlu ditegaskan bahwa penelitian ini difokuskan pada pengembangan dan pengujian fungsional/struktural sistem, sedangkan evaluasi performa kuantitatif model deteksi pelanggaran khusus (precision, recall, F1-score, mAP) masih merupakan tahap penelitian lanjutan yang belum tercakup dalam cakupan penelitian ini.

Metodologi

A. Desain Penelitian

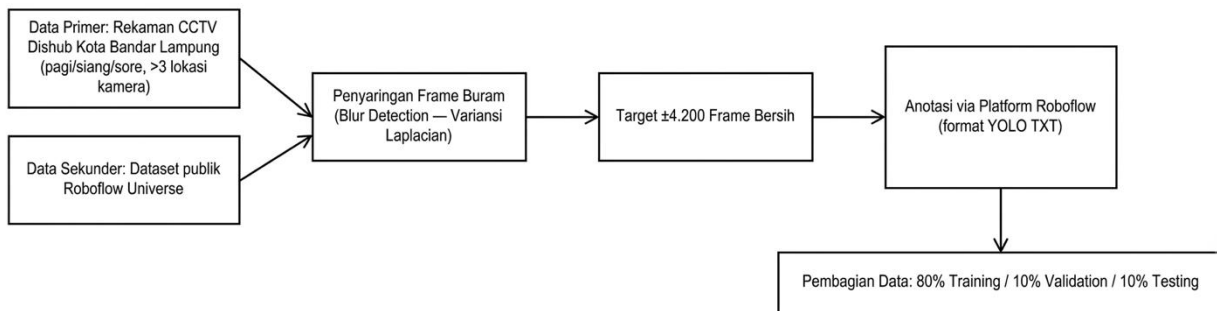
Penelitian ini menggunakan pendekatan development research (R&D) [14] dengan orientasi kuantitatif-eksperimental, yang diarahkan pada perancangan, implementasi, dan pengujian sebuah sistem monitoring lalu lintas berbasis machine learning. Tahapan penelitian mengikuti alur sebagai berikut: 1) identifikasi masalah dan studi literatur; 2) pengumpulan dan pengolahan data; 3) pengembangan dan pelatihan model YOLOv8; 4) perancangan dan implementasi sistem monitoring; 5) evaluasi model dan pengujian sistem; 6) analisis hasil dan penarikan kesimpulan. Perlu ditegaskan di awal bahwa tahapan 3) pada penelitian ini baru mencapai tahap pengembangan arsitektur sistem yang mendukung integrasi model, sementara pelatihan model khusus untuk deteksi empat kelas pelanggaran belum diselesaikan. Sistem saat ini beroperasi menggunakan bobot demo bawaan (yolov8n.pt, terlatih pada dataset COCO) sebagai placeholder fungsional, bukan model akhir yang dievaluasi performanya.



Gambar 1. Tahapan Penelitian Development Research (R&D)

B. Data

- 1) Sumber data. Data video/gambar lalu lintas dikumpulkan dari dua sumber: data primer berupa rekaman CCTV milik Dinas Perhubungan Kota Bandar Lampung, dan data sekunder dari dataset publik Roboflow Universe untuk melengkapi kelas pelanggaran yang sulit diperoleh secara mandiri dari rekaman CCTV.
- 2) Variasi pengambilan data. Pengumpulan data primer dilakukan pada tiga rentang waktu berbeda untuk mendapatkan variasi kondisi pencahayaan: pagi (06.00–09.00), siang (11.00–14.00), dan sore (16.00–19.00), diambil dari minimal tiga titik lokasi kamera yang berbeda.
- 3) Target volume data. Target minimal 4.200 frame bersih ditetapkan setelah proses penyaringan frame buram menggunakan deteksi blur berbasis variansi Laplacian.
- 4) Anotasi data. Anotasi kelas objek (kendaraan dan jenis pelanggaran) dilakukan melalui platform Roboflow [15], disimpan dalam format YOLO TXT.
- 5) Pembagian data. Data dibagi dengan proporsi 80% data latih (training), 10% data validasi (validation), dan 10% data uji (testing).



Gambar 2. Diagram Alur Pengumpulan dan Pengolahan Data

C. Perangkat Keras dan Perangkat Lunak

Pengembangan dan pengujian sistem dilakukan menggunakan perangkat keras berupa laptop dengan prosesor Intel Core i7-12700H/AMD Ryzen 7 5800H, GPU NVIDIA RTX 3060 6GB (mendukung CUDA 11.8 atau lebih baru), RAM 16GB DDR4, dan penyimpanan SSD NVMe 512GB.

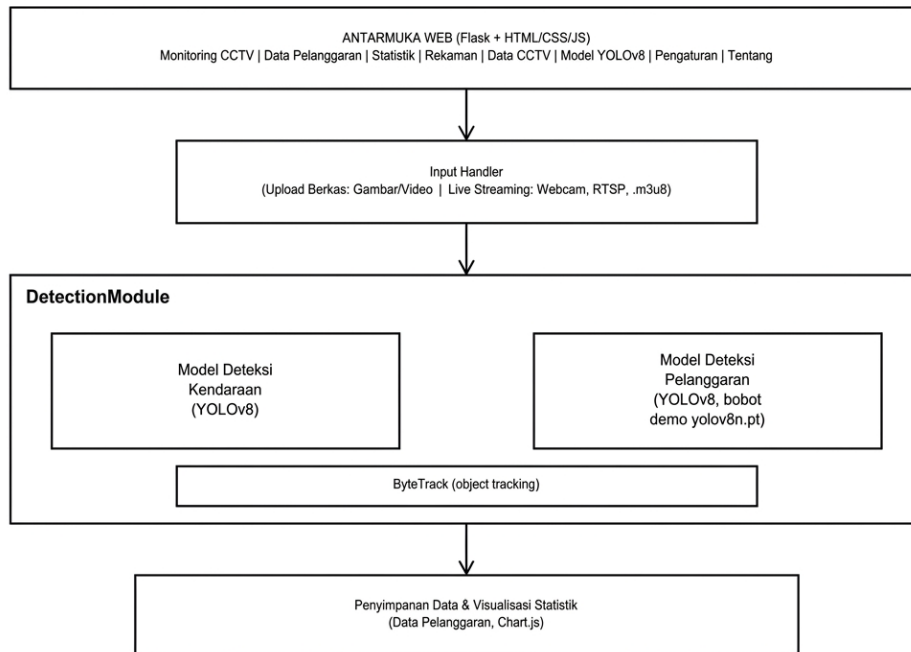
Pemilihan GPU pada tahap pelatihan model ditujukan untuk mengakselerasi proses pelatihan model deep learning, mengingat komputasi paralel GPU memberikan peningkatan signifikan pada kecepatan pelatihan dibandingkan CPU [16]. Perangkat lunak yang digunakan meliputi Python 3.10+, kerangka kerja Ultralytics YOLOv8 versi 8.x [17], PyTorch 2.x dengan dukungan CUDA, OpenCV 4.8+, kerangka kerja web Flask untuk backend aplikasi (menggantikan rencana awal penggunaan FastAPI), dan ByteTrack [18] untuk modul pelacakan objek (tracking), yang menggantikan rencana awal berbasis centroid tracking.

D. Perancangan dan Implementasi Sistem

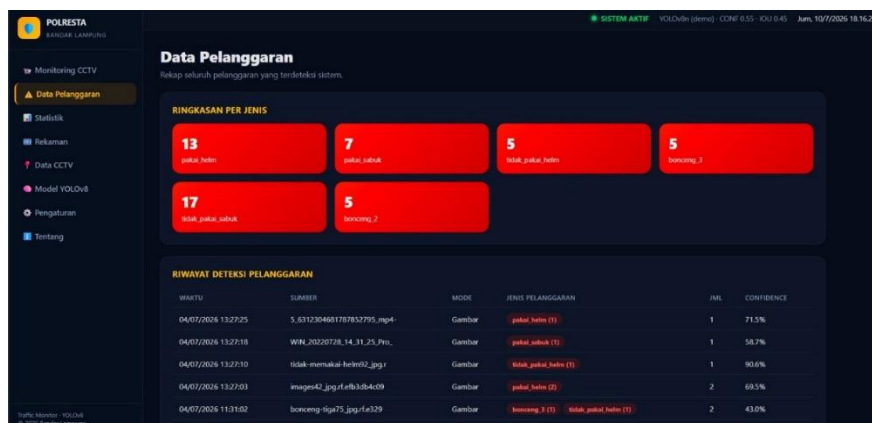
- 1) Arsitektur sistem. Sistem “Traffic Monitor” dirancang dengan arsitektur modular berbasis YOLOv8 [6] yang memisahkan secara eksplisit antara model deteksi kendaraan dan model deteksi pelanggaran, sehingga masing-masing modul dapat dikembangkan dan diperbarui secara independen.

2) Jalur input data. Sistem mendukung dua jalur input: berkas unggahan (gambar atau video), dan live streaming melalui webcam, protokol RTSP, atau tautan .m3u8.

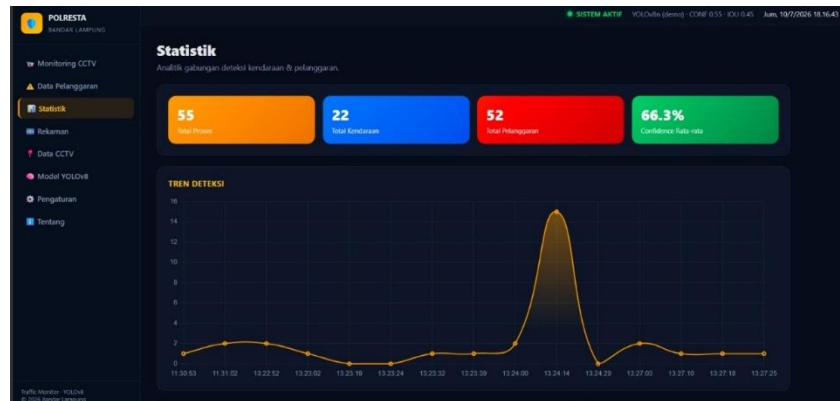
3) Antarmuka pengguna. Implementasi berupa aplikasi web berbasis Python dengan dashboard antarmuka HTML/CSS/JavaScript dan visualisasi data statistik menggunakan Chart.js. Dashboard terdiri atas delapan menu utama: Monitoring CCTV, Data Pelanggaran, Statistik, Rekaman, Data CCTV, Model YOLOv8, Pengaturan, dan Tentang.



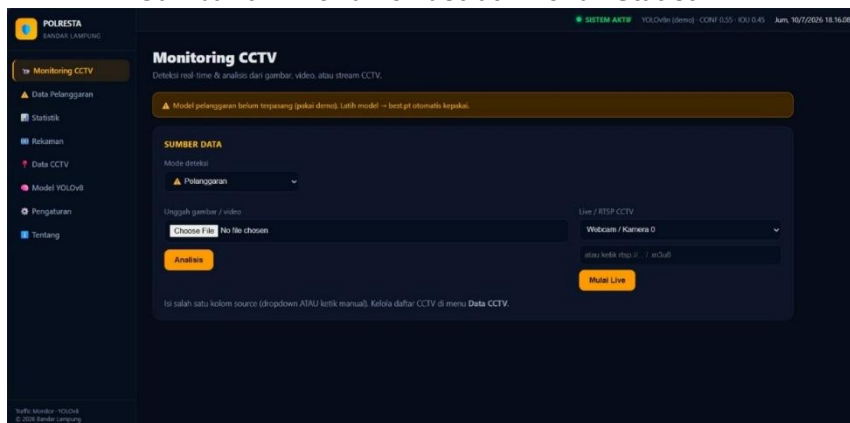
Gambar 3. Arsitektur Modular Sistem “Traffic Monitor”



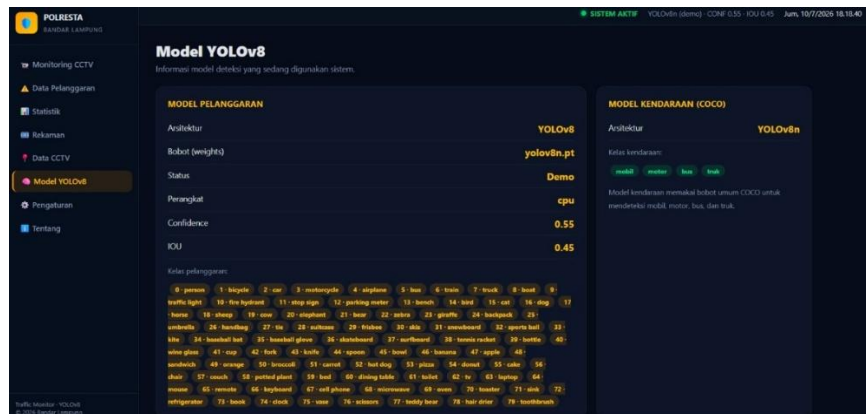
Gambar 5. Wireframe Ilustratif Menu “Data Pelanggaran”



Gambar 6. Wireframe Ilustratif Menu “Statistik”



Gambar 7. Wireframe Ilustratif Menu “Rekam”



Gambar 8. Wireframe Ilustratif Menu “Model YOLOv8”

4) Konfigurasi parameter. Beberapa parameter inferensi dapat dikonfigurasi langsung melalui antarmuka pengguna tanpa memerlukan modifikasi kode program, meliputi confidence threshold (diatur pada 0,55), IoU threshold (diatur pada 0,45), ukuran inferensi, dan frame stride.

License Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Doi: <http://dx.doi.org/xx.xxxxxx>
Vol 1 No 1 (July 2026)

5) Transparansi status model. Sebagai bagian dari prinsip perancangan yang bertanggung jawab, sistem menampilkan status model secara eksplisit kepada operator — termasuk indikasi bahwa model deteksi pelanggaran yang aktif saat ini masih bersifat demo — sehingga operator tidak salah menginterpretasikan keluaran sistem sebagai hasil model yang telah divalidasi penuh. Sistem ini ditujukan untuk digunakan oleh instansi terkait seperti Polresta Bandar Lampung dan Dinas Perhubungan Kota Bandar Lampung.

E. Pengujian dan Evaluasi Sistem

1) Pengujian fungsional (black-box testing) [19]. Pengujian ini dilakukan untuk memverifikasi bahwa perilaku sistem sesuai dengan skenario penggunaan yang ditetapkan, tanpa meninjau struktur kode internal. Skenario pengujian mencakup jalur unggah berkas, jalur live streaming, visualisasi hasil deteksi, penyimpanan data pelanggaran, serta navigasi antar-menu dashboard. Rincian skenario dan hasil pengujian disajikan pada Tabel 1.

Tabel 1. Skenario Pengujian Fungsional (Black-Box Testing)

No	Skenario Pengujian	Langkah Uji	Hasil Diharapkan	yang	Status
1	Unggah berkas gambar	Operator mengunggah berkas gambar melalui menu Monitoring CCTV	Sistem menerima berkas dan menjalankan inferensi deteksi		Sesuai harapan
2	Unggah berkas video	Operator mengunggah berkas video pada menu yang sama	Sistem memproses video frame-by-frame dan menampilkan hasil deteksi		Sesuai harapan
3	Live streaming via webcam	Operator memilih sumber webcam sebagai input	Sistem menampilkan feed video langsung dengan overlay hasil deteksi		Sesuai harapan
4	Live streaming via RTSP/.m3u8	Operator memasukkan URL RTSP/.m3u8 kamera CCTV	Sistem terhubung ke stream dan menjalankan deteksi real-time		Sesuai harapan
5	Konfigurasi parameter inferensi	Operator mengubah confidence/IoU threshold melalui menu Pengaturan	Perubahan parameter diterapkan tanpa modifikasi kode		Sesuai harapan
6	Penyimpanan data pelanggaran	Sistem mendeteksi objek pelanggaran pada input aktif	Data pelanggaran tersimpan dan tampil di menu Data Pelanggaran		Sesuai harapan
7	Penghitungan kendaraan (vehicle counting)	Sistem memproses video dengan objek bergerak	ByteTrack melacak dan menghitung kendaraan yang lewat		Sesuai harapan*
8	Visualisasi statistik	Operator membuka menu Statistik	Dashboard menampilkan grafik ringkasan data pelanggaran (Chart.js)		Sesuai harapan
9	Navigasi antar-menu dashboard	Operator berpindah antar 8 menu yang tersedia	Seluruh menu dapat diakses tanpa galat		Sesuai harapan
10	Transparansi status model	Operator membuka menu Model YOLOv8	Sistem menampilkan status model aktif		Sesuai harapan

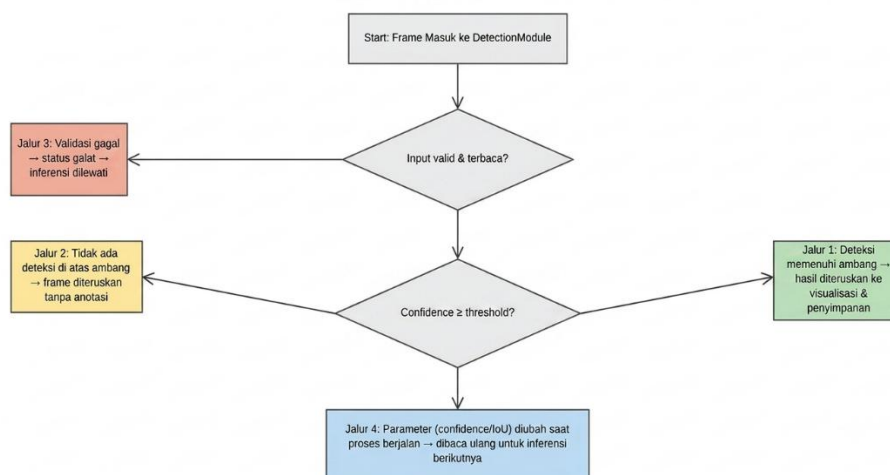
**Fungsi berjalan pada level alur kerja; akurasi kuantitatif vehicle counting belum diukur (lihat Keterbatasan pada Kesimpulan).*

2) Pengujian struktural (white-box testing) menggunakan analisis kompleksitas siklomatik [20]. Pengujian ini dilakukan pada modul DetectionModule untuk mengidentifikasi jalur eksekusi independen dalam logika modul. Hasil analisis menunjukkan nilai kompleksitas siklomatik yang tergolong rendah hingga sedang, dengan empat jalur independen teridentifikasi, sebagaimana disajikan pada Tabel 2. Pengujian ini didasarkan pada deskripsi logis modul dan belum dicocokkan secara langsung terhadap kode sumber aktual, sehingga hasil ini bersifat indikatif.

Tabel 2. Jalur Independen Hasil Analisis Kompleksitas Siklomatik — Modul DetectionModule

Jalur	Kondisi Percabangan Utama	Deskripsi Alur Eksekusi
Jalur 1	Input valid, objek terdeteksi di atas confidence threshold	Frame diproses → inferensi model → deteksi memenuhi ambang → hasil diteruskan ke modul visualisasi dan penyimpanan
Jalur 2	Input valid, tidak ada objek di atas confidence threshold	Frame diproses → inferensi model → tidak ada deteksi memenuhi ambang → frame diteruskan tanpa anotasi pelanggaran
Jalur 3	Input tidak valid/gagal dibaca (frame korup atau koneksi stream terputus)	Validasi input gagal → modul mengembalikan status galat → proses inferensi dilewati
Jalur 4	Perubahan parameter saat proses berjalan (confidence/loU threshold diubah di tengah sesi)	Parameter baru dibaca ulang oleh modul → inferensi berikutnya menggunakan nilai threshold terbaru

Kompleksitas siklomatik modul ini tergolong rendah hingga sedang, konsisten dengan jumlah jalur independen yang teridentifikasi (4 jalur), sehingga struktur logikanya relatif mudah dikelola dan diuji.



Gambar 9. Control-Flow Graph 4 Jalur Independen Modul DetectionModule

3) Evaluasi kesesuaian kebutuhan fungsional. Kesesuaian sistem terhadap enam kebutuhan fungsional yang ditetapkan pada tahap perancangan dievaluasi pada level antarmuka dan alur kerja, meliputi: streaming data video, deteksi objek, visualisasi hasil deteksi, penyimpanan data pelanggaran, penghitungan kendaraan (vehicle counting) melalui ByteTrack, dan dashboard statistik. Keenam

kebutuhan tersebut terpenuhi pada level fungsional/alur kerja, meskipun akurasi kuantitatif dari masing-masing fungsi berbasis model (deteksi pelanggaran spesifik dan vehicle counting) belum diukur secara terukur pada tahap penelitian ini.

Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sistem monitoring lalu lintas berbasis machine learning bernama "Traffic Monitor" yang mengintegrasikan deteksi empat jenis pelanggaran lalu lintas helm, bonceng tiga, sabuk pengaman, dan kecepatan berlebih dalam satu arsitektur sistem terpadu untuk konteks Kota Bandar Lampung. Sistem dibangun dengan arsitektur modular yang memisahkan model deteksi kendaraan dari model deteksi pelanggaran, mendukung dua jalur input (unggah berkas dan live streaming), serta menyediakan konfigurasi parameter inferensi yang dapat diatur langsung melalui antarmuka pengguna tanpa modifikasi kode.

Pengujian fungsional (black-box testing) menunjukkan bahwa alur kerja utama sistem mulai dari streaming data, deteksi objek, visualisasi, penyimpanan data pelanggaran, hingga navigasi dashboard berjalan sesuai skenario yang ditetapkan. Pengujian struktural (white-box testing) pada modul DetectionModule mengindikasikan kompleksitas siklomatik yang rendah hingga sedang dengan empat jalur independen teridentifikasi, menunjukkan struktur logika modul yang relatif mudah dikelola. Keenam kebutuhan fungsional sistem terpenuhi pada level antarmuka dan alur kerja. Dengan demikian, tujuan penelitian untuk merancang, mengimplementasikan, dan menguji secara fungsional dan struktural sistem monitoring lalu lintas berbasis YOLOv8 telah tercapai. Perlu ditegaskan bahwa penelitian ini memiliki keterbatasan yang signifikan dan disampaikan secara jujur sebagai bagian integral dari hasil: (1) model kustom untuk deteksi empat kelas pelanggaran belum selesai dilatih, sehingga sistem saat ini masih menggunakan bobot demo bawaan (yolov8n.pt) dan evaluasi kuantitatif performa model (precision, recall, F1-score, mAP) belum dapat disajikan; (2) deteksi pelanggaran kecepatan berlebih (overspeeding) belum terimplementasi; (3) akurasi penghitungan kendaraan (vehicle counting) melalui ByteTrack belum diuji secara terukur; (4) performa real-time sistem (FPS) belum diukur, dan inferensi saat ini masih berjalan berbasis CPU, bukan GPU; serta (5) pengujian struktural masih berbasis deskripsi logis modul dan belum dicocokkan langsung terhadap kode sumber aktual. Sebagai arah penelitian selanjutnya, disarankan untuk menyelesaikan pelatihan model kustom pada keempat kelas pelanggaran menggunakan dataset yang telah dikumpulkan, mengukur performa model secara kuantitatif dengan metrik standar (precision, recall, F1-score, mAP@50, mAP@50-95), mengimplementasikan modul deteksi overspeeding, serta melakukan pengujian performa real-time pada lingkungan inferensi berbasis GPU. Penyelesaian tahapan tersebut akan memungkinkan sistem dievaluasi secara lebih komprehensif sebagai solusi monitoring lalu lintas otomatis yang siap diterapkan pada skala operasional.

Daftar Pustaka

- [1] Korlantas Polri. (2024). Statistik kecelakaan lalu lintas. <https://korlantas.polri.go.id/statistik-laka/>
- [2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Dalam *Advances in Neural Information Processing Systems* (NeurIPS), 28.
- [3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. Dalam *Computer Vision – ECCV 2016* (pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2

- [4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779–788). <https://doi.org/10.1109/CVPR.2016.91>
- [5] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection (arXiv:2004.10934). arXiv. <https://doi.org/10.48550/arXiv.2004.10934>
- [6] Terven, J., Cordova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680–1716. <https://doi.org/10.3390/make5040083>
- [7] Neamah, S. B., & Karim, A. A. (2023). Real-time traffic monitoring system based on deep learning and YOLOv8. ARO-The Scientific Journal of Koya University, 11(2), 137–150. <https://doi.org/10.14500/aro.11327>
- [8] Shoman, M., Ghoul, T., Lanzaro, G., Alsharif, T., Gargoum, S., & Sayed, T. (2024). Enforcing traffic safety: A deep learning approach for detecting motorcyclists' helmet violations using YOLOv8 and deep convolutional generative adversarial network-generated images. Algorithms, 17(5), 202. <https://doi.org/10.3390/a17050202>
- [9] Sutikno, Sugiharto, A., & Kusumaningrum, R. (2023). Automated detection of driver and passenger without seat belt using YOLOv8. International Journal of Advanced Computer Science and Applications, 14(11). <https://doi.org/10.14569/IJACSA.2023.0141181>
- [10] Shaqib, S. M., Alo, A. P., Ramit, S. S., Rupak, A. U. H., Khan, S. S., & Rahman, M. S. (2024). Vehicle speed detection system utilizing YOLOv8: Enhancing road safety and traffic management for metropolitan areas (arXiv:2406.07710). arXiv. <https://doi.org/10.48550/arXiv.2406.07710>
- [11] Dou, H., Chen, S., Xu, F., Liu, Y., & Zhao, H. (2025). Analysis of vehicle and pedestrian detection effects of improved YOLOv8 model in drone-assisted urban traffic monitoring system. PLOS ONE, 20(3), e0314817. <https://doi.org/10.1371/journal.pone.0314817>
- [12] Dharmasaputra, K. D., & Hartato, B. P. (2025). Evaluation of YOLOv8 and centroid tracking in vehicle detection, classification, and counting system. Journal of Applied Informatics and Computing, 9(5), 2310–2319.
- [13] Premaratne, P., Kadhim, I. J., Blacklidge, R., & Lee, M. (2023). Comprehensive review on vehicle detection, classification and counting on highways. Neurocomputing, 556, 126627. <https://doi.org/10.1016/j.neucom.2023.126627>
- [14] Borg, W. R., & Gall, M. D. (1983). Educational research: An introduction (4th ed.). Longman.
- [15] Ciaglia, F., Zuppichini, F. S., Guerrie, P., McQuade, M., & Solawetz, J. (2022). Roboflow 100: A rich, multi-domain object detection benchmark (arXiv:2211.13523). arXiv. <https://doi.org/10.48550/arXiv.2211.13523>
- [16] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
- [17] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLOv8 by Ultralytics (Versi 8.0.0) [Perangkat lunak]. Ultralytics. <https://github.com/ultralytics/ultralytics>
- [18] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). ByteTrack: Multi-object tracking by associating every detection box. Dalam Computer Vision – ECCV 2022 (pp. 1–21). Springer. https://doi.org/10.1007/978-3-031-20047-2_1
- [19] Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
- [20] McCabe, T. J. (1976). A complexity measure. IEEE Transactions on Software Engineering, SE-2(4), 308–320. <https://doi.org/10.1109/TSE.1976.233837>