ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

SISTEM PENDETEKSI LAMPU LALU LINTAS UNTUK KENDARAAN PINTAR MENGGUNAKAN YOLOv9 BERBASIS PYTORCH

Achmad Dewa✉, Ni Gusti Putu Ayu Harry Saptarini, I Putu Astya Prayudha
Teknologi Informasi, Politeknik Negeri Bali, Badung, Indonesia
Email: achmaddewa15@gmail.com

DOI: https://doi.org/10.46880/jmika.Vol10No2.pp543-553

ABSTRACT

The ability to accurately detect traffic lights in real time is a crucial component for the safety and navigation of autonomous vehicles. This study aims to develop and evaluate a traffic light detection system (red, yellow, green) using the YOLOv9 deep learning algorithm based on the PyTorch framework. This study employs a quantitative approach with an experimental method utilizing the S2TLD dataset as well as additional covering various lighting and weather conditions. Unlike previous methods that rely on computationally heavy architectural modifications, this study highlights the novelty of utilizing the native YOLOv9 architecture to efficiently resolve information bottlenecks on small objects. The model was evaluated using Precision, Recall, Mean Average Precision (mAP), and inference time metrics. Test results show that the YOLOv9 model is highly effective, achieving a 98.0% mAP@0.5 in standalone testing and 96.5% when tested directly on a dashboard system using a local dataset. Additionally, the system recorded 0% cross-color classification errors on the test dataset, demonstrating a high level of reliability to support smart vehicle systems. In terms of processing speed, the system achieves an average inference time of 41.1 ms per frame (±24 FPS) using GPU acceleration. In conclusion, the YOLOv9 architecture has proven to be accurate, safe, and robust for implementation as an autonomous navigation perception system.
Keyword: Autonomous Vehicles, Traffic Lights, YOLOv9, Computer Vision, Inference Time.

ABSTRAK

Kemampuan mendeteksi lampu lalu lintas secara akurat dan real-time merupakan komponen krusial bagi keselamatan dan navigasi kendaraan otonom. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi lampu lalu lintas (merah, kuning, hijau) menggunakan algoritma deep learning YOLOv9 berbasis framework PyTorch. Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen yang memanfaatkan dataset S2TLD serta data tambahan yang mencakup berbagai variasi kondisi pencahayaan dan cuaca. Berbeda dengan metode sebelumnya yang bergantung pada modifikasi arsitektur yang berat secara komputasi, penelitian ini menonjolkan kebaruan melalui pemanfaatan arsitektur native YOLOv9 yang efisien untuk memecahkan fenomena information bottleneck pada objek kecil. Model dievaluasi menggunakan metrik Precision, Recall, Mean Average Precision (mAP), dan waktu inferensi. Hasil pengujian menunjukkan bahwa model YOLOv9 sangat efektif dengan capaian mAP@0.5 sebesar 98,0% pada pengujian mandiri dan 96,5% saat diuji langsung pada sistem dashboard menggunakan dataset lokal. Selain itu, sistem mencatat 0% kesalahan klasifikasi silang antar-warna pada dataset pengujian sehingga menunjukkan tingkat keandalan yang tinggi untuk mendukung sistem kendaraan pintar. Dari segi kecepatan pemrosesan, sistem mampu mencapai rata-rata waktu inferensi sebesar 41,1 ms per frame (±24 FPS) menggunakan akselerasi GPU. Kesimpulannya, arsitektur YOLOv9 terbukti akurat, aman, dan tangguh untuk diimplementasikan sebagai sistem persepsi navigasi mandiri.
Kata Kunci: Kendaraan Otonom. Lampu Lalu Lintas, YOLOv9, Computer Vision, Waktu Inferensi.

PENDAHULUAN

Perkembangan teknologi kendaraan otonom dalam beberapa tahun terakhir menunjukkan kemajuan yang signifikan seiring meningkatnya kebutuhan akan sistem transportasi yang aman, efisien, dan cerdas. Kendaraan otonom memanfaatkan berbagai teknologi seperti sensor, *Global Positioning System* (GPS), dan kecerdasan buatan *(Artificial Intelligence)* untuk melakukan navigasi serta pengambilan keputusan secara mandiri tanpa intervensi manusia. Salah satu kemampuan penting yang harus dimiliki kendaraan otonom adalah memahami dan mematuhi aturan lalu lintas, termasuk mengenali kondisi lampu lalu lintas secara akurat dan real-time (Liao et al., 2025).

Deteksi lampu lalu lintas merupakan komponen krusial dalam sistem kendaraan otonom karena informasi yang diperoleh digunakan sebagai dasar pengambilan keputusan, seperti berhenti,

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

memperlambat kendaraan, atau melanjutkan perjalanan. Kesalahan dalam mendeteksi status lampu lalu lintas dapat menyebabkan pelanggaran aturan lalu lintas maupun kecelakaan yang membahayakan pengguna jalan. Namun, proses deteksi lampu lalu lintas menghadapi berbagai tantangan, seperti perubahan intensitas pencahayaan, kondisi cuaca yang tidak menentu, ukuran objek yang relatif kecil, serta keberadaan berbagai objek lain yang dapat mengganggu proses identifikasi (Golcarenarenji et al., 2024).

Perkembangan teknologi computer vision dan deep learning telah memberikan solusi yang menjanjikan dalam bidang deteksi objek. Salah satu algoritma yang banyak digunakan adalah *You Only Look Once* (YOLO), yang mampu melakukan deteksi objek secara cepat dan akurat dalam satu tahap pemrosesan.

Dibandingkan metode deteksi objek konvensional, YOLO memiliki keunggulan dalam kecepatan inferensi sehingga sangat sesuai untuk aplikasi *real-time* seperti kendaraan otonom. Namun, penelitian terdahulu yang menggunakan iterasi YOLO sebelumnya sering kali dihadapkan pada fenomena *information bottleneck*, di mana informasi penting dari objek berukuran kecil (seperti lampu lalu lintas dari kejauhan) hilang saat melewati ekstraksi fitur yang dalam. Untuk mengatasi kelemahan tersebut, studi-studi sebelumnya terpaksa melakukan modifikasi arsitektur yang ekstensif dan kompleks. Sebagai contoh, model KCS-YOLO yang dibangun dari arsitektur dasar YOLOv5n harus menyematkan mekanisme atensi CBAM (*Convolutional Block Attention Module*) dan menambahkan *small-target detection layer* secara manual untuk meningkatkan akurasi pada visibilitas rendah. Demikian pula pada iterasi YOLOv8, model seperti TSD-YOLO mengandalkan penambahan modul *Space-to-Depth* (SPD) dan atensi *Select Kernel* (SK) untuk mempertahankan fitur rambu berskala kecil, sementara model AW-YOLO memerlukan penambahan D-DWR dan *DySample* untuk beradaptasi dengan kondisi cuaca buruk.(Wang et al., 2024; Wu & Fan, 2025)

Dari studi-studi terdahulu, masih terdapat celah penelitian (*research gap*) mengenai belum adanya sistem persepsi yang mampu mempertahankan fitur objek rambu berskala kecil pada jarak jauh tanpa memperbesar beban komputasi. Oleh karena itu, alasan utama pemilihan YOLOv9 pada penelitian ini dibandingkan versi YOLO sebelumnya didasari oleh kebaruan arsitektur *native*-nya yang secara inheren mengintegrasikan *Programmable Gradient Information* (PGI) dan *Generalized Efficient Layer Aggregation Network* (GELAN). Pendekatan PGI mampu mengatasi masalah hilangnya fitur spasial (*information bottleneck*) dengan mempertahankan informasi input secara utuh untuk menghitung fungsi kerugian, sehingga menghasilkan pembaruan bobot yang sangat akurat. Keunggulannya, YOLOv9 mampu mengekstraksi semantik tingkat tinggi secara efisien tanpa memerlukan *overhead* komputasi dari penjejalan modul arsitektur tambahan—seperti modifikasi SPD atau mekanisme atensi (CBAM/SK)—yang selama ini menjadi kelemahan mendasar pada model-model generasi sebelumnya.

Beberapa penelitian sebelumnya telah menunjukkan keberhasilan penerapan YOLO dalam deteksi lampu lalu lintas. Penelitian terdahulu melaporkan bahwa model berbasis YOLO mampu mendeteksi lampu lalu lintas dengan tingkat presisi yang tinggi pada berbagai kondisi lingkungan(Zhou et al., 2024). Selain itu, pendekatan deep learning yang dikombinasikan dengan teknik pemrosesan citra modern terbukti dapat meningkatkan kemampuan deteksi pada kondisi pencahayaan rendah maupun lingkungan perkotaan yang kompleks(Possatti et al., 2019). Hasil penelitian lainnya juga menunjukkan bahwa algoritma YOLO memiliki kemampuan deteksi multiobjek yang baik tanpa mengorbankan kecepatan pemrosesan, sehingga berpotensi untuk diimplementasikan pada sistem kendaraan cerdas yang membutuhkan respons cepat (Flores-Calero et al., 2024).

Meskipun demikian, evaluasi performa YOLOv9 untuk deteksi lampu lalu lintas pada berbagai kondisi lingkungan masih memerlukan kajian lebih lanjut. Oleh karena itu, penelitian ini berfokus pada pengembangan dan evaluasi model YOLOv9 untuk mendeteksi lampu lalu lintas merah, kuning, dan hijau secara real-time. Dataset yang digunakan berasal dari sumber terbuka, seperti S2TLD (*Small-Scale Traffic Light Dataset*) dan dataset pendukung lainnya yang mencakup variasi kondisi pencahayaan dan cuaca. Proses penelitian dilakukan melalui tahapan pengumpulan dan persiapan dataset, pelatihan model YOLOv9, serta pengujian performa menggunakan metrik evaluasi seperti mean Average Precision (mAP) dan waktu inferensi.

Tujuan penelitian ini adalah mengembangkan sistem deteksi lampu lalu lintas berbasis YOLOv9 yang mampu mendeteksi objek secara cepat dan akurat, mengevaluasi kinerja sistem pada berbagai kondisi lingkungan, serta menganalisis kemampuan model dalam mendukung kebutuhan deteksi real-time. Hasil penelitian diharapkan dapat memberikan kontribusi terhadap pengembangan teknologi computer vision

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

pada kendaraan otonom serta menjadi referensi bagi penelitian lanjutan dalam bidang transportasi cerdas dan sistem deteksi objek berbasis deep learning.

Kebaharuan dari penelitian ini terletak pada pemanfaatan kemampuan inheren arsitektur YOLOv9 untuk memecahkan fenomena *information bottleneck* pada objek lalu lintas berskala kecil, serta validasinya menggunakan kondisi operasional jalan raya lokal yang unik. Oleh karena itu, penelitian ini merumuskan tiga kontribusi ilmiah utama:

1. Validasi empiris performa arsitektur *Programmable Gradient Information* (PGI) dan *Generalized Efficient Layer Aggregation Network* (GELAN) pada YOLOv9 di luar batasan dataset umum (seperti S2TLD), khususnya terhadap variabilitas visual dataset lalu lintas lokal.
2. Analisis kritis terhadap keselamatan sistem (*safety analysis*) kendaraan otonom berdasarkan evaluasi tingkat kesalahan fatal klasifikasi silang antar-warna (*fatal cross-class misclassification*).
3. Pembuktian efisiensi komputasi melalui sistem *dashboard* deteksi *real-time* yang memvalidasi kesiapan model untuk diimplementasikan pada perangkat *edge* kendaraan pintar.

KAJIAN LITERATUR

Sistem Persepsi Kendaraan Otonom

Perkembangan teknologi kendaraan pintar (*smart vehicle*) dan kendaraan otonom telah mendorong kebutuhan akan sistem persepsi yang mampu memahami lingkungan secara akurat dan *real-time*. Salah satu komponen penting dalam sistem persepsi tersebut adalah kemampuan mendeteksi dan mengenali lampu lalu lintas sebagai dasar pengambilan keputusan kendaraan dalam menjalankan manuver seperti berhenti, melambat, maupun melanjutkan perjalanan. Ketepatan deteksi lampu lalu lintas menjadi faktor krusial karena berhubungan langsung dengan keselamatan dan kepatuhan kendaraan terhadap aturan lalu lintas (Chen et al., 2025).

Computer Vision

Untuk memenuhi kebutuhan sistem persepsi tersebut, teknologi *computer vision* banyak digunakan dalam sistem kendaraan otonom. *Computer vision* adalah cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memampukan mesin atau komputer untuk memperoleh, memproses, menganalisis, dan memahami informasi visual dari citra digital maupun video di dunia nyata. Dalam konteks deteksi lampu lalu lintas, *computer vision* berfungsi untuk mengekstraksi fitur visual guna mengidentifikasi lokasi (berupa *bounding box*), bentuk, dan warna lampu lalu lintas berdasarkan piksel data visual yang diperoleh dari kamera sensor kendaraan.

Pendekatan Deep Learning

Implementasi *computer vision* pada deteksi objek saat ini sangat didominasi oleh pendekatan *deep learning*. *Deep learning* memungkinkan model mempelajari representasi fitur secara otomatis dari data dalam jumlah besar melalui jaringan saraf tiruan berlapis (*artificial neural networks*), tanpa memerlukan ekstraksi fitur secara manual oleh manusia. Kemampuan komputasi hierarkis tersebut menjadikan *deep learning* sangat adaptif terhadap berbagai variasi kondisi lingkungan, seperti perubahan intensitas pencahayaan (siang/malam), cuaca, variasi jarak objek, maupun perubahan sudut pengambilan gambar.

Algoritma You Only Look Once (YOLO)

Salah satu algoritma *deep learning* yang paling revolusioner dan banyak digunakan untuk deteksi objek adalah *You Only Look Once* (YOLO). Berbeda dengan metode deteksi konvensional yang memisahkan tahap pencarian lokasi objek dan klasifikasi kelas (*two-stage detector*), YOLO merumuskan deteksi objek sebagai masalah regresi tunggal (*single-stage detector*) (Bakirci, 2024).

YOLO bekerja dengan membagi gambar masukan ke dalam ukuran *grid* sebesar . Setiap sel *grid* bertanggung jawab memprediksi kotak pembatas (*bounding boxes*), tingkat keyakinan (*confidence score*), dan probabilitas kelas objek secara simultan dalam satu kali proses komputasi maju (*forward pass*). Hal ini membuat YOLO sangat unggul dalam kecepatan inferensi, sehingga menjadikannya algoritma standar untuk aplikasi yang membutuhkan respons *real-time* seperti pada sistem navigasi kendaraan otonom (Öztürk et al., 2024).

Arsitektur YOLOv9

Perkembangan iteratif algoritma tersebut saat ini telah menghasilkan YOLOv9. Versi ini mengintegrasikan pendekatan arsitektur baru, yaitu *Programmable Gradient Information* (PGI) dan *Generalized Efficient Layer Aggregation Network* (GELAN) (Xie et al., 2025). Arsitektur PGI mengatasi masalah hilangnya informasi pada *deep neural networks* selama proses transmisi antar *layer*, sementara GELAN dirancang untuk meningkatkan efisiensi penggunaan parameter komputasi sekaligus mempertahankan kualitas representasi fitur. Penelitian Wang et al. (2024) menunjukkan bahwa YOLOv9 memiliki keunggulan performa yang signifikan, baik dalam aspek akurasi maupun efisiensi komputasi,

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

dibandingkan model *state-of-the-art* lainnya seperti YOLOv7, YOLOv8, RT-DETR, maupun PP-YOLOE pada pengujian set data MS COCO (Assagaf et al., 2026.)

Metrik Evaluasi Performa Model

Untuk mengukur sejauh mana keandalan model *deep learning* dalam mendeteksi objek, evaluasi performa dilakukan menggunakan metrik standar yang diturunkan dari *Confusion Matrix*. Parameter dasar yang digunakan meliputi *True Positive* (TP) yaitu prediksi benar target objek terdeteksi, *False Positive* (FP) yaitu prediksi keliru di mana latar belakang dianggap sebagai objek, dan *False Negative* (FN) yaitu kondisi di mana target objek gagal terdeteksi oleh sistem.

Precision

Precision digunakan untuk mengukur rasio tingkat ketepatan prediksi model. Metrik ini menunjukkan persentase jumlah deteksi yang benar-benar relevan (objek lampu lalu lintas) dibandingkan dengan keseluruhan deteksi positif yang dihasilkan model.

$$\boldsymbol{Precision = \frac{TP}{TP + FP}}$$

Recall

Recall menunjukkan kemampuan model dalam menemukan seluruh target objek aktual yang seharusnya terdeteksi di dalam citra. Semakin tinggi nilai *recall*, semakin sedikit objek yang terlewatkan (FN) oleh sistem.

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}$$

F1-Score

F1-Score merupakan rata-rata harmonik (*harmonic mean*) dari *Precision* dan *Recall*. Metrik ini sangat berguna untuk menemukan keseimbangan antara presisi dan sensitivitas, terutama ketika terdapat distribusi data yang tidak seimbang (*imbalanced dataset*).

$$\boldsymbol{F1 - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}}$$

Mean Average Precision (mAP)

mAP merupakan metrik evaluasi utama dan paling komprehensif dalam deteksi objek. Metrik ini dihitung melalui luas area di bawah Kurva *Precision-Recall* (AUC-PR) yang disebut sebagai *Average Precision* (AP) untuk satu kelas, yang kemudian dirata-ratakan untuk seluruh jumlah kelas (N).

$$\boldsymbol{mAP = \frac{1}{N} \backslash sum_{i=1}^{N} AP_i}$$

Penelitian Terdahulu

Penelitian mengenai deteksi lampu lalu lintas berbasis deep learning telah menunjukkan perkembangan yang signifikan dalam beberapa tahun terakhir. (Wang et al., 2024) melaporkan bahwa YOLOv9 mampu meningkatkan akurasi dan efisiensi parameter dibandingkan berbagai model deteksi objek sebelumnya. Namun, penelitian tersebut masih berfokus pada dataset umum dan belum mengevaluasi performanya secara khusus pada domain deteksi lampu lalu lintas.

Zhou et al. (2024) mengembangkan model KCS-YOLO berbasis YOLOv5n untuk mendeteksi lampu lalu lintas pada kondisi visibilitas rendah. Penelitian tersebut berhasil mencapai nilai mAP sebesar 98,87%, tetapi pengujian masih terbatas pada dataset kustom dengan jumlah data yang relatif kecil serta belum mencakup variasi kondisi lingkungan yang lebih kompleks.

(Possatti et al., 2019) mengombinasikan YOLOv3 dengan prior maps dan sensor LiDAR untuk meningkatkan relevansi deteksi lampu lalu lintas pada kendaraan otonom. Meskipun mampu meningkatkan akurasi deteksi, pendekatan tersebut memerlukan arsitektur sistem yang kompleks dan ketergantungan terhadap sensor tambahan, sehingga implementasinya menjadi kurang praktis.

Selain itu, (Flores-Calero et al., 2024) melalui studi literatur sistematis terhadap 115 penelitian menemukan bahwa algoritma YOLO memiliki keunggulan dalam aspek kecepatan inferensi dan akurasi untuk aplikasi kendaraan otonom. Akan tetapi, penelitian tersebut tidak melakukan eksperimen langsung terhadap model yang dikaji sehingga belum memberikan gambaran empiris mengenai performa algoritma pada kondisi operasional nyata

Tabel 1. Peneletian Sebelumnya

Penelitian	Metode dan Dataset	Hasil	Kelemahan	Posisi Penelitian
Wang et al. (2024)	Metode: YOLOv9 (PGI & GELAN)	Akurasi (AP) dan efisiensi meningkat	Pengujian masih berfokus pada dataset	Mengadopsi YOLOv9 secara

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

	Dataset: MS COCO	dibanding YOLOv7, YOLOv8, RT-DETR dan PP-YOLOE	umum, belum diuji spesifik pada domain objek tertentu	khusus untuk domain deteksi lampu lalu lintas pada kondisi jalan dunia nyata
Zhou et al. (2024)	Metode:KSC-YOLO Dataset: 3.053 Citra kustom	Mencapai mAP 98,87; unggul dalam mendeteksi lampu pada kondisi kabut,hujan dan malam	Belum mempertimbangkan kondisi ektrem lai (sialu matahari, refleksi) dan pengujian terbatas	Menggunakan arsitektur model yang lebih baru dan fokus pada eksperimen inferensi real-time
Possatti et al. (2019)	Metode: YOLOv3 dipadu prior maps dan LiDAR Dataset:DTLD,LISA-TLD, & IARA-TLD	Sistem akurat mengidentifikasi lampu relevan dengan deteksi awal pada jarak hingga ±90 meter	Arsitektur sangat kompleks karena bergantung pada sensor tambahan dan prior maps	Mengevaluasi performa model AI murni tanpa ketergantungan prior maps atau sensor tambahan
Flores-Calero et al. (2024)	Metode: Systemic Literature Review (SLR) Dataset: 115 studi primer (2016-2022)	Algoritma YOLO terbukti sangat unggul untuk kecapatan inferensi real-time dan akurasi ADAS	Hanya kajian literatur tanpa eksperimen langsung, serta terbatas pada YOLO versi lama (v1-v5)	Melanjutkan temuan SLR tersebut dengan melakukan eksperimen langsung menggunakan YOLO terbaru

Berdasarkan kajian terhadap penelitian terdahulu, masih terdapat celah penelitian berupa kebutuhan evaluasi empiris terhadap performa YOLOv9 dalam mendeteksi lampu lalu lintas pada berbagai kondisi lingkungan nyata dengan tetap mempertahankan kecepatan inferensi yang memenuhi kebutuhan sistem real-time. Selain itu, kajian mengenai kontribusi model terhadap pengambilan keputusan pada kendaraan pintar masih relatif terbatas.

METODE PENELITIAN

Penelitian ini merupakan penelitian terapan (*applied research*) yang bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi lampu lalu lintas berbasis kecerdasan buatan menggunakan algoritma YOLOv9. Fokus penelitian adalah menganalisis kemampuan model dalam mendeteksi dan mengklasifikasikan lampu lalu lintas berwarna merah, kuning, dan hijau secara real-time untuk mendukung sistem persepsi pada kendaraan pintar.

Pendekatan penelitian yang digunakan adalah pendekatan kuantitatif dengan metode eksperimen. Pendekatan kuantitatif dipilih karena kinerja model dievaluasi menggunakan parameter yang dapat diukur secara objektif, seperti *Precision*, *Recall*, *Mean Average Precision* (mAP), serta waktu inferensi. Metode eksperimen digunakan untuk menguji kemampuan model dalam mendeteksi lampu lalu lintas pada berbagai kondisi lingkungan, meliputi perbedaan pencahayaan, kondisi cuaca, dan variasi posisi objek.

Objek penelitian berupa sistem deteksi lampu lalu lintas yang dibangun menggunakan algoritma YOLOv9 berbasis framework PyTorch. Sistem dirancang untuk mengenali tiga kelas lampu lalu lintas, yaitu merah, kuning, dan hijau. Dataset penelitian diperoleh dari sumber terbuka berupa S2TLD (*Small-Scale Traffic Light Dataset*), serta data tambahan yang dikumpulkan secara mandiri untuk memperkaya variasi kondisi lingkungan. Secara keseluruhan, dataset ini berjumlah 2.740 citra. Mengacu pada standar evaluasi model deteksi objek yang objektif, dataset dibagi menjadi tiga bagian utama dengan rasio 80:10:10, yaitu: data *training* sebesar 80%, data *validation* sebesar 10%, dan data *testing* sebesar 10%. Distribusi kelas di dalam dataset ini juga dijaga keseimbangannya, yang terdiri dari 816 anotasi untuk kelas lampu merah, 465 anotasi untuk kelas lampu kuning, dan 1.459 anotasi untuk kelas lampu hijau. Dataset terdiri atas citra dan video yang merepresentasikan kondisi siang hari, malam hari, cuaca cerah, dan cuaca hujan.

Tahapan penelitian diawali dengan proses akuisisi data, yaitu pengumpulan dataset lampu lalu lintas dari berbagai sumber. Selanjutnya dilakukan tahap pra-pemrosesan data yang meliputi ekstraksi frame video menggunakan OpenCV, validasi data, penyesuaian ukuran citra menjadi 640 × 640 piksel, normalisasi nilai piksel, serta augmentasi data berupa rotasi, *flipping*, perubahan tingkat kecerahan, dan variasi skala objek. Tahapan ini bertujuan untuk meningkatkan kualitas data dan kemampuan generalisasi model. Proses anotasi (*labeling*) dilakukan secara manual untuk memetakan *bounding box* ke dalam tiga kelas utama (merah, kuning, dan hijau) dengan format standar YOLO. Mengacu pada pedoman

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

evaluasi model deteksi objek yang objektif, dataset yang berjumlah total 2.740 citra ini dipartisi menggunakan rasio 80:10:10, yang mendistribusikan 80% data untuk *training*, 10% untuk *validation*, dan 10% untuk *testing*.

Setelah proses pelabelan dan pra-pemrosesan selesai, dilakukan pelatihan model menggunakan arsitektur YOLOv9c selama 50 *epoch* melalui *framework* PyTorch (Ultralytics). Proses komputasi memanfaatkan akselerasi perangkat keras berbasis *cloud* menggunakan GPU NVIDIA Tesla T4 15GB. Adapun *hyperparameter* yang ditetapkan dalam tahap pelatihan ini meliputi resolusi citra masukan sebesar 640x640 piksel dan *batch size* sebanyak 16. Untuk mengoptimalkan pembaruan bobot (*weights*) selama pelatihan, digunakan fungsi optimasi AdamW dengan *learning rate* sebesar 0,001429, tingkat momentum 0,9, dan *weight decay* sebesar 0,0005. Selain itu, guna meningkatkan kemampuan generalisasi model dan mencegah *overfitting*, diterapkan strategi augmentasi data otomatis yang mencakup *mosaic* (1.0), *random erasing* (0.4), *horizontal flip* (0.5), serta variasi transformasi warna dan filter (seperti *blur* dan CLAHE). Hasil pelatihan ini kemudian digunakan pada tahap inferensi untuk mendeteksi keberadaan lampu lalu lintas dan mengklasifikasikannya ke dalam kategori merah, kuning, atau hijau.

Pengumpulan data dilakukan melalui observasi terhadap hasil deteksi model pada dataset pengujian yang tidak digunakan selama proses pelatihan. Skenario pengujian dirancang untuk merepresentasikan kondisi operasional kendaraan pintar, meliputi kondisi pencahayaan siang hari, malam hari, cuaca cerah, cuaca hujan atau mendung, variasi jarak objek terhadap kamera, serta variasi posisi lampu lalu lintas pada persimpangan jalan.

Analisis kinerja model dilakukan menggunakan beberapa metrik evaluasi, yaitu *Precision*, *Recall*, *Mean Average Precision* (mAP@0.5), *confusion matrix*, dan waktu inferensi. Precision digunakan untuk mengukur tingkat ketepatan prediksi model, Recall digunakan untuk mengukur kemampuan model dalam menemukan objek yang seharusnya terdeteksi, sedangkan mAP digunakan untuk mengevaluasi performa keseluruhan model deteksi objek. Selain itu, waktu inferensi digunakan untuk menilai kemampuan sistem dalam memenuhi kebutuhan deteksi secara real-time. Hasil evaluasi kemudian dianalisis untuk mengetahui efektivitas algoritma YOLOv9 dalam mendeteksi lampu lalu lintas pada berbagai kondisi lingkungan dan mendukung implementasi kendaraan pintar.

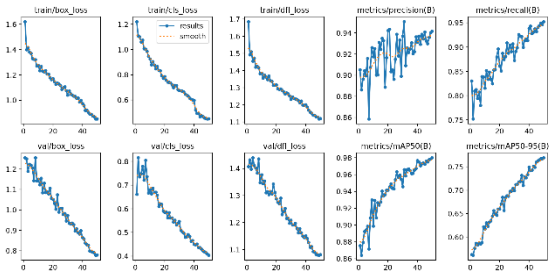


Gambar 1. Alur Penelitian

HASIL PENELITIAN

Pelatihan Model

Pelatihan model YOLOv9 dilakukan selama 50 *epoch* untuk mengoptimalkan kemampuan pengenalan fitur lampu lalu lintas dari *dataset*.



Gambar 2. Grafik Metrik Pelatihan dan Validasi Model Yolov9 Selama 50 Epoch

Berdasarkan Gambar 2, terlihat grafik train/box_loss, train/cls_loss, dan train/dfl_loss mengalami penurunan yang stabil sejak epoch pertama hingga epoch ke-50. Penurunan yang konsisten pada fungsi kerugian (loss function) ini mengindikasikan bahwa model berhasil mempelajari fitur spasial (lokasi bounding box) dan klasifikasi objek dengan baik tanpa

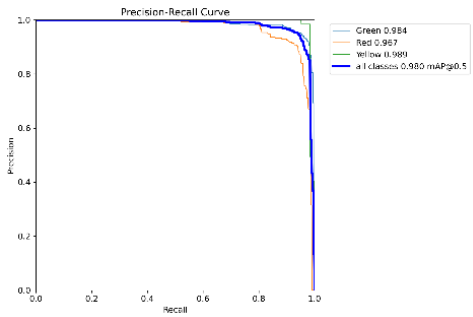
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

mengalami underfitting. Sejalan dengan hal tersebut, grafik val/box_loss, val/cls_loss, dan val/dfl_loss juga menunjukkan pola penurunan yang serupa, membuktikan bahwa model memiliki kemampuan generalisasi yang baik pada data yang belum pernah dilihat sebelumnya (unseen data) dan tidak mengalami overfitting.Metrik kinerja utama seperti Precision, Recall, mAP@0.5, dan mAP@0.5:0.95 menunjukkan lonjakan peningkatan di awal pelatihan dan mencapai stabilitas di epoch akhir. Nilai mAP@0.5 berhasil mencapai angka di kisaran 0.98, yang membuktikan tingginya tingkat keandalan model YOLOv9 dalam melakukan deteksi.

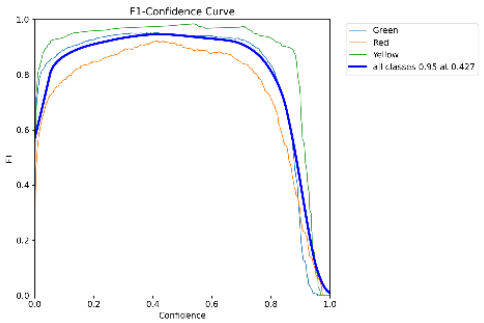
Evaluasi Metrik Kineja

Untuk mengevaluasi keseimbangan antara ketepatan (*precision*) dan sensitivitas (*recall*) model dalam mendeteksi tiap kelas lampu lalu lintas, digunakan kurva *Precision-Recall* (PR *Curve*).



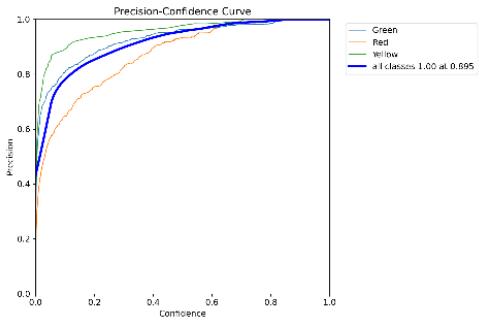
Gambar 3. Kurva Precision-Recall (PR Curve)

Kurva PR pada Gambar 3 menunjukkan performa keseluruhan model mencapai nilai mAP@0.5 sebesar 0.980. Jika dirinci berdasarkan kelas, performa terbaik dicapai pada deteksi lampu kuning (*Yellow*) dengan mAP 0.989, disusul lampu hijau (*Green*) 0.984, dan lampu merah (*Red*) 0.967. Area di bawah kurva (AUC) yang mendekati nilai 1.0 merepresentasikan model dapat mempertahankan tingkat presisi yang sangat tinggi meskipun tingkat *recall* (kemampuan menemukan seluruh objek) ditingkatkan.



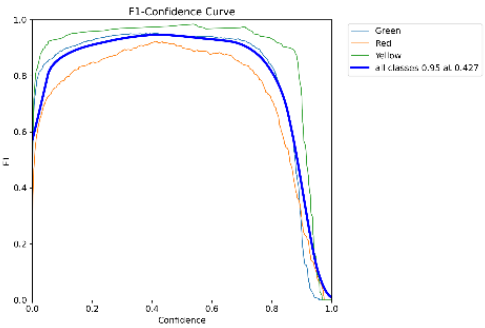
Gambar 4. Kurva F1-Score

Metrik kinerja utama seperti Precision, Recall, mAP@0.5, dan mAP@0.5:0.95 menunjukkan lonjakan peningkatan di awal pelatihan dan mencapai stabilitas di epoch akhir. Nilai mAP@0.5 berhasil mencapai angka di kisaran 0.98, yang membuktikan tingginya tingkat keandalan model YOLOv9 dalam melakukan deteksi.



Gambar 5. Kurva Precision-Confidence

Kurva *Precision-Confidence* (Gambar 5) mengilustrasikan hubungan antara tingkat kepercayaan (*confidence threshold*) prediksi model dengan tingkat presisi yang dihasilkan. Berdasarkan kurva tersebut, metrik presisi untuk seluruh kelas (*all classes*) mencapai nilai sempurna (1.00) pada ambang batas kepercayaan 0.895. Hal ini mengindikasikan bahwa apabila sistem mendeteksi sebuah objek lampu lalu lintas dengan tingkat keyakinan 89,5% atau lebih, maka menunjukkan tingkat akurasi yang sangat tinggi pada data pengujian.



Gambar 6. Kurva Recall-Confidence

Di sisi lain, Kurva *Recall-Confidence* (Gambar 6) menyoroti seberapa baik kemampuan model dalam menemukan seluruh target objek yang sebenarnya (*ground truth*) di berbagai tingkat kepercayaan. Pada tingkat kepercayaan awal atau terendah (0.000), model berhasil mencatatkan nilai *Recall* sebesar 0.99. Ini membuktikan bahwa model memiliki sensitivitas yang sangat tinggi dan mampu mendeteksi 99% dari total lampu lalu lintas yang ada di dalam *dataset* pengujian, nyaris tidak melewatkan objek satupun. Seiring dengan

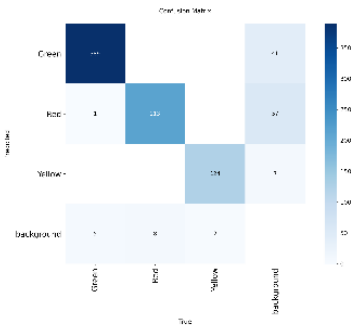
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

dinaikkannya ambang batas kepercayaan untuk menyaring deteksi, nilai *recall* perlahan menurun secara wajar.

Analisis Confusion Matrix

Evaluasi lebih rinci terhadap kesalahan klasifikasi (*misclassification*) dianalisis menggunakan *Confusion Matrix*, baik dalam nilai absolut maupun ternormalisasi.

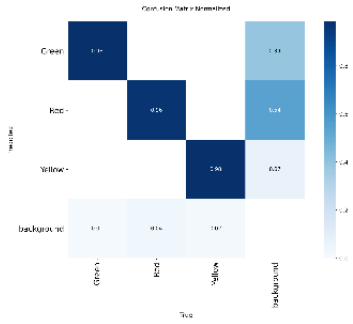


Gambar 7. Confusion Matrix (Absolut)

Evaluasi lebih rinci terhadap jumlah prediksi dan penyebaran klasifikasinya disajikan melalui *Confusion Matrix* Absolut (Gambar 7). Nilai yang berada pada garis diagonal utama merepresentasikan *True Positives* (prediksi benar), di mana model terbukti sukses mendeteksi dan mengklasifikasikan 389 lampu hijau (*Green*), 213 lampu merah (*Red*), dan 124 lampu kuning (*Yellow*) dengan akurat.

Confusion Matrix membedah beberapa batasan model dalam mendeteksi objek. Pada metrik *False Positives*, terlihat ada area latar belakang yang keliru dikenali sebagai lampu lalu lintas: 41 dideteksi sebagai hijau, 57 sebagai merah, dan 7 sebagai kuning. Kesalahan deteksi ini umumnya dipicu oleh latar belakang lingkungan urban yang kompleks (*complex backgrounds*), seperti kemiripan fitur dengan papan reklame neon, lampu jalan, atau pepohonan yang menutupi. Di sisi lain, jumlah *False Negatives* sangat minim, yaitu hanya gagal mendeteksi 5 lampu hijau, 8 merah, dan 2 kuning. Kegagalan deteksi ini (*False Negatives*) terjadi akibat jarak yang sangat jauh serta visibilitas rendah yang mengaburkan warna asli objek ke dalam latar belakang atmosfer.

Satu temuan penting dari matriks absolut ini adalah sangat minimnya kesalahan klasifikasi silang antar-kelas (*cross-class misclassification*). Dari seluruh pengujian, hanya terdapat 1 insiden di mana lampu hijau terdeteksi sebagai lampu merah, sedangkan kesalahan identifikasi antar-warna lainnya tidak terjadi sama sekali (bernilai 0). Ini menunjukkan sistem memiliki akurasi yang tinggi karena tidak kebingungan dalam membedakan antara sinyal berhenti (merah) dan melaju (hijau).



Gambar 8. Confusion Matrix (Ternomalisasi)

Berdasarkan Gambar 8 (*Confusion Matrix* Ternormalisasi), algoritma YOLOv9 memiliki tingkat akurasi klasifikasi (*True Positive Rate*) yang sangat superior. Model mampu mengklasifikasikan kelas *Green* (Hijau) dengan akurasi 98%, *Yellow* (Kuning) 98%, dan *Red* (Merah) 96%. Rasio kebingungan dengan latar belakang (*background false positives/negatives*) terpantau sangat minim, di mana hanya 1-4% kesalahan prediksi terjadi karena objek lampu lalu lintas tidak terdeteksi atau latar belakang dikenali sebagai lampu. Model juga sama sekali tidak menunjukkan kesalahan silang yang fatal antarkelas warna (misalnya, lampu merah yang diprediksi sebagai hijau nilainya adalah 0.00).

Secara analitis, ketiadaan kesalahan klasifikasi silang antar-warna yang fatal (0% *fatal cross-color misclassification*) memiliki implikasi krusial terhadap standar keselamatan (*safety standards*) kendaraan otonom. Dalam skenario operasional nyata di jalan raya Indonesia yang dinamis dan kerap memiliki pencahayaan tidak ideal, keandalan identifikasi ini mencegah kendaraan mengambil keputusan *stop-and-go* yang membahayakan lalu lintas sekitar. Capaian waktu inferensi sebesar 41,1 ms per *frame* (±24 FPS) turut membuktikan bahwa sistem persepsi ini tidak mengalami latensi, sehingga mampu memberikan umpan balik visual seketika (*real-time response*) yang vital bagi unit kontrol manuver kendaraan.

Keberhasilan YOLOv9 dalam mempertahankan fitur objek rambu berskala kecil pada jarak jauh menegaskan bahwa mekanisme *Programmable Gradient Information* (PGI) secara efektif mengatasi hilangnya informasi (*information bottleneck*) pada jaringan dalam. Temuan empiris ini secara langsung menjawab tantangan struktural yang pada penelitian sebelumnya harus disiasati dengan cara yang boros komputasi. Sebagai perbandingan, (Zhou et al., 2024) harus menambahkan mekanisme atensi secara manual

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

pada YOLOv5 untuk mendeteksi lampu pada visibilitas rendah, sementara (Possatti et al., 2019) harus bergantung pada integrasi kompleks antara sensor LiDAR dan peta *a priori*. Analisis ini mengonfirmasi bahwa arsitektur *native* YOLOv9 terbukti efisien dan tangguh sebagai sistem persepsi tunggal berbasis penglihatan (*vision-based*) tanpa memerlukan penambahan perangkat keras eksternal.

PEMBAHASAN PENELITIAN

Perbandingan dengan Model Deteksi Lain

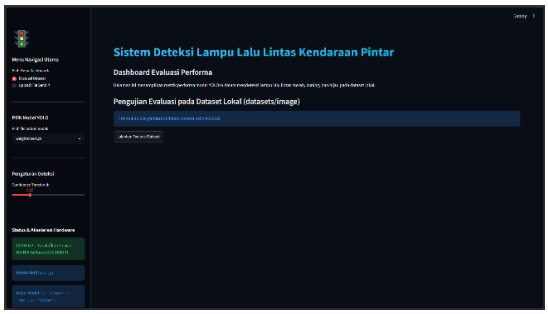
Untuk membuktikan keunggulan empiris arsitektur yang diusulkan, performa YOLOv9 dibandingkan secara kuantitatif dengan literatur model arsitektur deteksi objek lainnya. Berdasarkan hasil eksperimen, YOLOv9 mencapai nilai mAP@0.5 sebesar 98,0%. Nilai ini mengungguli model *single-stage* populer seperti YOLOv8s yang umumnya berada di kisaran akurasi 83,3%(Du et al., 2024), YOLOv5s di kisaran 81,5%(Öztürk et al., 2024), dan arsitektur SSD yang hanya mencapai 54,2% pada dataset lalu lintas. Lebih lanjut, YOLOv9 juga jauh mengungguli model *two-stage* seperti Faster R-CNN, yang diketahui sering mengalami kesulitan dalam mengekstraksi fitur objek kecil secara utuh pada jarak jauh, di mana Faster R-CNN hanya mencatatkan mAP sebesar 24,31% untuk deteksi lampu lalu lintas.

Analisis Kinerja Berdasarkan Kondisi Lingkungan

Kemampuan adaptasi model YOLOv9 juga dievaluasi terhadap variasi kondisi lingkungan spesifik, mengingat gangguan cuaca dapat menghilangkan fitur objek yang menonjol dan menurunkan iluminasi lingkungan. Pada pengujian skenario siang hari dengan visibilitas optimal, model beroperasi dengan performa maksimal. Pada skenario dengan visibilitas rendah seperti malam hari, kondisi hujan, mendung, atau silau, jarak objek yang jauh dapat menurunkan kontras kecerahan antara lampu lalu lintas dan latar belakangnya. Meski demikian, augmentasi data selama pelatihan (seperti variasi *blur*, transformasi warna, dan *random erasing*) dipadukan dengan arsitektur GELAN terbukti mampu mempertahankan ekstraksi fitur objek, menjadikan arsitektur ini tangguh sebagai sistem persepsi visual.

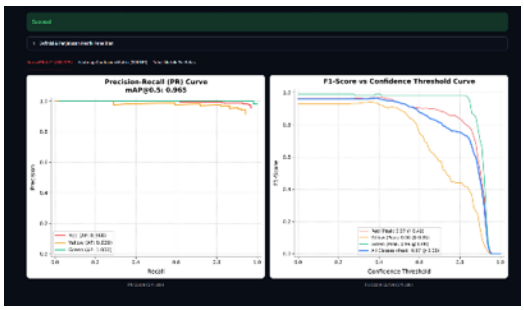
Implementasi Sistem dan Evaluasi Dataset Lokal

Model YOLOv9 yang telah dilatih (*best.pt*) kemudian diintegrasikan ke dalam sebuah sistem berbasis *web dashboard* untuk memudahkan proses pengujian dan evaluasi lebih lanjut. Sistem ini dijalankan dengan dukungan akselerasi perangkat keras (*hardware acceleration*) menggunakan GPU NVIDIA GeForce GTX 1650 Ti dan arsitektur CUDA untuk memastikan komputasi dapat berjalan secepat mungkin.



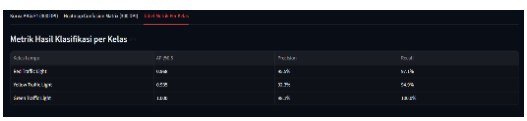
Gambar 9. Antarmuka Utama

Melalui sistem ini, dilakukan pengujian evaluasi ulang menggunakan *dataset* validasi lokal yang terdiri dari 276 gambar. Pengujian ini bertujuan untuk melihat performa model secara langsung di dalam lingkungan sistem yang telah dibangun.



Gambar 10. Kurva Precision – Recall dan F1-Score pada Pengujian Sistem

Berdasarkan hasil evaluasi sistem (Gambar 10), model mempertahankan performa yang sangat baik dengan mencatatkan nilai rata-rata mAP@0.5 sebesar 0.965. Puncak keseimbangan performa (*F1-Score*) untuk keseluruhan kelas mencapai 0.97 pada ambang batas kepercayaan (*confidence threshold*) 0.38.



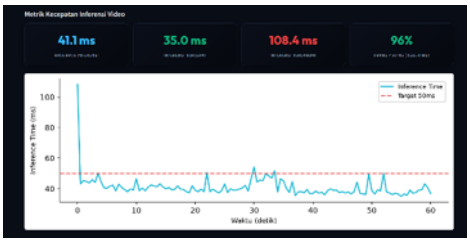
Gambar 11. Metrik Hasil Tabel Sistem

Rincian metrik per kelas (Gambar 11) menunjukkan bahwa lampu Hijau (*Green Traffic Light*) memiliki tingkat deteksi paling sempurna dengan *Average Precision* (AP) 1.000, *Precision* 98.1%, dan *Recall* 100.0%. Lampu Merah (*Red*) mencatat AP 0.968, dan Lampu Kuning (*Yellow*) mencatat AP 0.935. Hasil ini membuktikan bahwa model sangat andal dalam mengenali lampu hijau yang krusial untuk

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

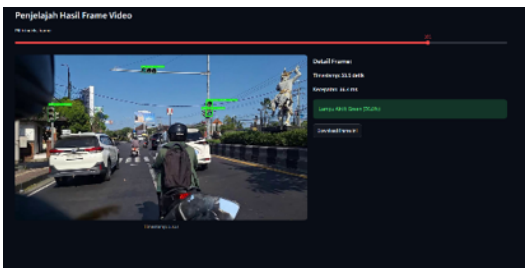
ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

keputusan kendaraan melaju, serta sangat baik dalam mengenali lampu merah untuk perintah berhenti.



Gambar 12. Metrik Kecepatan Inferensi Video pada Sistem

Hasil pengujian kecepatan inferensi (Gambar 12) menunjukkan bahwa sistem ini sangat memenuhi kualifikasi *real-time*. Waktu inferensi rata-rata tercatat sebesar 41,1 milidetik (ms) per *frame*, dengan rekor waktu inferensi tercepat menyentuh 35,0 ms. Dari total durasi pengujian selama 60 detik, sebanyak 96% *frame* berhasil diproses di bawah target waktu 50 ms. Kecepatan rata-rata 41,1 ms ini setara dengan pemrosesan sekitar 24 *Frame Per Second* (FPS), yang sangat ideal untuk penglihatan komputer pada kendaraan yang sedang bergerak.



Gambar 13. Visualisai Deteksi Lampu Lalu Lintas pada Ekstraksi Frame Video Lalu Lintas Nyata

Gambar 13 memperlihatkan bukti visual keberhasilan sistem saat mendeteksi lampu hijau (*Green*) dengan tingkat kepercayaan 76,8% pada kondisi jalan raya di siang hari yang cukup padat. Model mampu secara presisi mengotakkan (*bounding box*) lampu lalu lintas meskipun ukurannya relatif kecil dari kejauhan dan terdapat berbagai gangguan visual (seperti tiang listrik, baliho, dan kendaraan lain) di sekitarnya.

KESIMPULAN

Penelitian ini memberikan beberapa kontribusi penting dalam pengembangan sistem persepsi kendaraan pintar, antara lain:

1. Mengembangkan model deteksi lampu lalu lintas berbasis YOLOv9.
2. Mengevaluasi performa model menggunakan dataset S2TLD dan dataset lokal.
3. Mengimpelemtasikan model ke dalam dashboard berbasis web untuk pengujian real-time.

Penelitian ini berhasil mengembangkan sistem deteksi lampu lalu lintas berbasis YOLOv9 menggunakan *framework* PyTorch untuk mendukung kendaraan pintar. Hasil pengujian menunjukkan bahwa model mampu mencapai nilai mAP@0.5 sebesar 98,0% pada pengujian model dan 96,5% pada pengujian sistem menggunakan dataset lokal. Sistem juga mampu melakukan deteksi secara real-time dengan rata-rata waktu inferensi 41,1 ms per frame atau sekitar 24 FPS. Hasil tersebut menunjukkan bahwa YOLOv9 memiliki performa yang baik dalam mendeteksi dan mengklasifikasikan lampu lalu lintas merah, kuning, dan hijau sehingga berpotensi diterapkan pada sistem persepsi kendaraan pintar.

Namun demikian, penelitian ini memiliki sejumlah keterbatasan. Evaluasi pada sistem dashboard saat ini masih bergantung pada penggunaan dataset lokal yang belum mencakup berbagai wilayah geografis yang lebih luas, di mana variabilitas bentuk dan standar infrastruktur jalan dapat berbeda. Selain itu, belum ada pengujian komprehensif menggunakan dataset yang secara khusus dirancang untuk kondisi cuaca sangat ekstrem seperti kabut tebal atau badai salju (seperti dataset ACDC) yang sering memicu noise visual yang masif. Sebagai potensi pengembangan lanjutan, disarankan untuk memperluas cakupan dataset pada skenario cuaca ekstrem serta mengimplementasikan model ini secara langsung pada perangkat komputasi *edge* di kendaraan otonom fisik untuk memvalidasi kinerja inferensi dan latensi secara lebih komprehensif.

DAFTAR PUSTAKA

Assagaf, I. P. A., & Prasetiyo, A. B. (2026). Kecerdasan Buatan dalam Kendaraan Otonom: Tinjauan Sistematis terhadap Tren dan Tantangan Masa Depan. Journal of Energy, Materials, & Manufacturing Technology, 5(01), 40-48. https://doi.org/10.61844/jemmtec.v5i01.1366

Bakirci, M. (2024). Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems. *Traitement Du Signal*, *41*(04), 1727–1740. https://doi.org/10.18280/ts.410407

Chen, D., Dai, L., Yue, X., Gu, Q., Huang, S., Pan, J., Zhang, Y., & He, M. (2025). AW-YOLO: A Multi-Object Detection Network for Autonomous Driving Under all Weather Conditions. *IET Image Processing*, *19*(1). https://doi.org/10.1049/ipr2.70111

Du, S., Pan, W., Li, N., Dai, S., Xu, B., Liu, H., Xu, C., & Li, X. (2024). TSD-YOLO: Small traffic

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

sign detection based on improved YOLO v8. *IET Image Processing*, *18*(11), 2884–2898. https://doi.org/10.1049/ipr2.13141

Flores-Calero, M., Astudillo, C. A., Guevara, D., Maza, J., Lita, B. S., Defaz, B., Ante, J. S., Zabala-Blanco, D., & Armingol Moreno, J. M. (2024). Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. In *Mathematics* (Vol. 12, Number 2). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/math12020297

Golcarenarenji, G., Martinez-Alpiste, I., Wang, Q., & Alcaraz-Calero, J. M. (2024). Robust real-time traffic light detector on small-form platform for autonomous vehicles. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, *28*(5), 668–678. https://doi.org/10.1080/15472450.2023.2205018

Liao, J., Jiang, S., Chen, M., & Sun, C. (2025). SAM-YOLO: An Improved Small Object Detection Model for Vehicle Detection. *The European Journal on Artificial Intelligence*, *38*(3), 279–295. https://doi.org/10.1177/30504554251319452

Öztürk, G., Eldoğan, O., & Köker, R. (2024). Computer Vision-Based Lane Detection and Detection of Vehicle, Traffic Sign, Pedestrian Using YOLOv5. *Sakarya University Journal of Science*, *28*(2), 418–430. https://doi.org/10.16984/saufenbilder.1393307

Possatti, L. C., Guidolini, R., Cardoso, V. B., Berriel, R. F., Paixão, T. M., Badue, C., De Souza, A. F., & Oliveira-Santos, T. (2019). *Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars*. https://doi.org/10.1109/IJCNN.2019.8851927

Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*. http://arxiv.org/abs/2402.13616

Wu, T., & Fan, X. (2025). Nonlinear Filtering and YOLOv9-Based Moving Object Tracking Method Under Dynamic Scenarios. *Journal of Circuits, Systems and Computers*, *34*(18), 2550203. https://doi.org/10.1142/S0218126625502032

Xie, Y., Du, D., & Liu, Y. (2025). RE-YOLO: Vehicle Detector Based on Knowledge Distillation YOLOv8 for Autonomous Driving Scenarios. *Journal of Computing in Civil Engineering*, *39*(6), 4025084. https://doi.org/10.1061/JCCEE5.CPENG-6164

Zhou, Q., Zhang, D., Liu, H., & He, Y. (2024). KCS-YOLO: An Improved Algorithm for Traffic Light Detection under Low Visibility Conditions. *Machines 2024, Vol. 12, Page 557*, *12*(8), 557. https://doi.org/10.3390/MACHINES12080557

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.