Sistem Pengaturan Lampu Lalu Lintas Adaptif Berbasis Deteksi Volume Kendaraan Menggunakan YOLOv8 dan Raspberry Pi

1st Muhammad Althariq
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
muhammadalthariq@student.telkomuniversity.ac.id

2nd Raden Roro Dinda Ayu Pramesti
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
dindayps@student.telkomuniversity.ac.id

3rd Rendy Munadi
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
rendymunadi@telkomuniversity.ac.id

4th Sofia Naning Hertiana
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
sofiananing@telkomuniversity.ac.id

***Abstrak*—Kemacetan pada persimpangan bersinyal masih menjadi permasalahan penting dalam sistem transportasi perkotaan karena sistem lampu lalu lintas berbasis waktu tetap belum mampu menyesuaikan perubahan volume kendaraan secara *real-time*. Penelitian ini merancang prototipe sistem pengaturan lampu lalu lintas adaptif berbasis deteksi volume kendaraan menggunakan YOLOv8 dan Raspberry Pi 5. Sistem menggunakan video CCTV untuk mendeteksi kendaraan, menghitung jumlah kendaraan, mengonversinya ke Satuan Mobil Penumpang (SMP), menghitung arus lalu lintas dalam SMP/jam, serta menentukan derajat kejenuhan (DS). Nilai DS digunakan sebagai dasar penentuan durasi lampu hijau adaptif, sedangkan lampu kuning ditetapkan selama 3 detik. Hasil simulasi lima skenario menunjukkan bahwa sistem adaptif mampu menurunkan rata-rata antrean dari 141,66 menjadi 95,96 kendaraan dan waktu tunggu dari 167,24 menjadi 104,44 detik.**

Kata kunci: lampu lalu lintas adaptif, YOLOv8, Raspberry Pi, *computer vision*, derajat kejenuhan, *vehicle counting*.

I. PENDAHULUAN

Permasalahan kemacetan pada persimpangan bersinyal merupakan isu yang terus meningkat seiring dengan pertumbuhan jumlah kendaraan dan mobilitas masyarakat perkotaan. Persimpangan menjadi titik kritis karena mempertemukan beberapa arus kendaraan dalam satu ruang konflik. Ketidaksesuaian antara volume kendaraan dan durasi lampu lalu lintas dapat menyebabkan antrean, tundaan, pemborosan bahan bakar, serta penurunan efisiensi jaringan jalan [1], [2]. Pada banyak persimpangan, pengaturan lampu masih menggunakan sistem *fixed-time control*, yaitu durasi merah, kuning, dan hijau ditentukan berdasarkan waktu tetap atau data historis. Sistem ini sederhana, tetapi kurang responsif terhadap fluktuasi lalu lintas aktual [3], [4].

Sistem pengaturan lampu lalu lintas adaptif dikembangkan untuk menjawab keterbatasan tersebut. Sistem adaptif mampu menyesuaikan durasi lampu berdasarkan data lalu lintas aktual sehingga alokasi waktu hijau dapat diberikan lebih proporsional terhadap arah dengan beban kendaraan lebih tinggi [5], [6]. Sebagian sistem adaptif konvensional menggunakan sensor fisik seperti *loop detector* atau sensor magnetik. Meskipun mampu menyediakan data lalu lintas, sensor fisik memiliki keterbatasan berupa biaya instalasi, ketergantungan pada perangkat jalan, cakupan deteksi terbatas, dan potensi kerusakan fisik [7], [8].

Perkembangan *computer vision* dan *deep learning* membuka peluang pemanfaatan kamera CCTV sebagai sensor visual lalu lintas. Kamera yang telah tersedia di banyak persimpangan dapat digunakan untuk mendeteksi dan menghitung kendaraan tanpa pemasangan sensor tambahan pada permukaan jalan. Model deteksi objek berbasis YOLO, khususnya YOLOv8, banyak digunakan karena mampu melakukan deteksi secara cepat dan akurat pada citra lalu lintas yang kompleks [9], [10], [11], [12]. Akan tetapi, banyak penelitian masih berfokus pada deteksi dan penghitungan kendaraan, sementara integrasi hasil deteksi ke dalam mekanisme keputusan durasi lampu hijau adaptif masih perlu dikembangkan lebih lanjut [13], [14].

Berdasarkan permasalahan tersebut, penelitian ini merancang sistem pengaturan lampu lalu lintas adaptif berbasis deteksi volume kendaraan menggunakan YOLOv8 dan Raspberry Pi. Kontribusi utama penelitian ini adalah integrasi *end-to-end* antara deteksi kendaraan, *vehicle counting*, konversi SMP, perhitungan Q SMP/jam, perhitungan DS, penentuan durasi hijau adaptif, *dashboard monitoring*, dan prototipe *traffic light* berbasis Raspberry Pi. Sistem tidak dimaksudkan untuk menggantikan APILL lapangan secara langsung, melainkan sebagai prototipe dan sistem pendukung keputusan yang tetap mempertahankan urutan serta makna isyarat lampu lalu lintas sesuai ketentuan yang berlaku [15], [16], [17].

II. TINJAUAN PUSTAKA SINGKAT

A. Pengaturan Lampu Lalu Lintas Adaptif

Pengaturan lampu lalu lintas adaptif atau *adaptive traffic signal control* merupakan pendekatan pengaturan sinyal yang menyesuaikan waktu lampu berdasarkan kondisi lalu lintas aktual. Berbeda dengan *fixed-time control*, sistem adaptif dapat mengubah durasi hijau sesuai volume atau kepadatan kendaraan yang terdeteksi. Pendekatan ini dinilai mampu meningkatkan jumlah kendaraan terlayani serta menurunkan antrean dan waktu tunggu pada kondisi lalu lintas yang dinamis [5], [6]. Dalam konteks penelitian ini, adaptivitas

diterapkan pada durasi lampu hijau, sedangkan durasi lampu kuning dibuat tetap sebagai waktu transisi keselamatan.

B. Deteksi Kendaraan Menggunakan YOLOv8

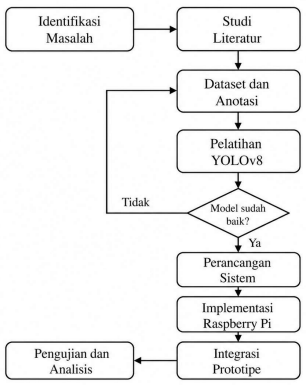
YOLOv8 merupakan algoritma deteksi objek satu tahap yang melakukan prediksi lokasi dan kelas objek secara langsung dalam satu proses inferensi. Karakteristik ini membuat YOLOv8 sesuai untuk aplikasi *real-time* seperti pemantauan lalu lintas. Studi komparatif menunjukkan bahwa YOLOv8 memiliki keseimbangan baik antara akurasi dan kecepatan inferensi dibandingkan beberapa arsitektur deteksi lain seperti Faster R-CNN dan EfficientDet [18], [19]. Dalam penelitian ini, YOLOv8 digunakan untuk mendeteksi tiga kelas kendaraan, yaitu *car*, *motorcycle*, dan *heavy vehicle*.

C. SMP, Q SMP/jam, dan Derajat Kejenuhan

Analisis lalu lintas memerlukan penyetaraan berbagai jenis kendaraan karena setiap kelas kendaraan memiliki pengaruh berbeda terhadap kapasitas jalan. Oleh karena itu, jumlah kendaraan dikonversi ke dalam satuan mobil penumpang (SMP). Nilai arus lalu lintas dalam SMP/jam selanjutnya dibandingkan dengan kapasitas jalan untuk memperoleh derajat kejenuhan (DS). DS digunakan sebagai indikator tingkat pemanfaatan kapasitas jalan dan menjadi dasar penentuan kategori volume lalu lintas [20], [21]. Pada penelitian ini, nilai DS digunakan untuk menentukan durasi lampu hijau adaptif.

III. METODOLOGI PENELITIAN

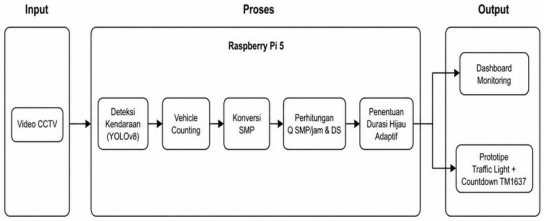
Penelitian ini menggunakan pendekatan perancangan, implementasi, dan pengujian prototipe. Alur penelitian dimulai dari identifikasi masalah, studi literatur, penyusunan dataset, pelatihan YOLOv8, perancangan sistem, implementasi pada Raspberry Pi, integrasi prototipe, serta pengujian dan analisis. Diagram alur penelitian ditunjukkan pada Gambar 1.



GAMBAR 1
Diagram alur penelitian sistem pengaturan lampu lalu lintas adaptif.

A. Arsitektur Kerja Sistem

Sistem menerima video CCTV sebagai masukan. Frame video diproses menggunakan model YOLOv8 untuk mendeteksi kendaraan. Hasil deteksi digunakan untuk menghitung jumlah kendaraan pada interval pengamatan tertentu. Jumlah kendaraan kemudian dikonversi ke SMP, dihitung sebagai Q SMP/jam, dan dibandingkan dengan kapasitas jalan untuk memperoleh nilai DS. Nilai DS digunakan untuk menentukan durasi hijau adaptif. Keluaran sistem ditampilkan pada *dashboard monitoring* dan direpresentasikan pada prototipe *traffic light* dengan *countdown* TM1637. Diagram kerja sistem ditunjukkan pada Gambar 2.



GAMBAR 2
Diagram kerja sistem pengaturan lampu lalu lintas adaptif berbasis deteksi kendaraan.

B. Dataset dan Pelatihan Model

Dataset kendaraan terdiri atas tiga kelas, yaitu *car*, *motorcycle*, dan *heavy vehicle*. Dataset dikembangkan dalam empat versi pelatihan, yaitu Pelatihan 1, Pelatihan 2, Pelatihan 3, dan Pelatihan 4. Pembagian data dilakukan menggunakan komposisi 70% *training*, 20% *validation*, dan 10% *testing*. Ringkasan jumlah dataset ditunjukkan pada Tabel 1.

TABEL 1
Jumlah dataset pada setiap versi pelatihan YOLOv8.

Versi	Training	Validation	Testing	Total
Pelatihan 1	700	200	100	1000
Pelatihan 2	1400	400	200	2000
Pelatihan 3	2100	600	300	3000
Pelatihan 4	2240	640	320	3200

Model YOLOv8 dilatih menggunakan *pretrained weight* YOLOv8n. Parameter utama pelatihan meliputi 50 *epoch*, ukuran citra 640 piksel, dan *batch size* 8. Evaluasi model dilakukan menggunakan metrik *precision*, *recall*, *F1-score*, dan *mAP@0.5*. Pemilihan model final tidak hanya didasarkan pada nilai metrik, tetapi juga mempertimbangkan kestabilan hasil deteksi pada video CCTV kondisi siang dan malam.

C. Perhitungan Volume, Q SMP/jam, dan DS

Jumlah kendaraan yang terdeteksi pada interval pengamatan dinyatakan sebagai volume kendaraan. Secara umum, volume kendaraan dihitung menggunakan Persamaan (1).

$$Q = \frac{N}{T} \tag{1}$$

Pada Persamaan (1), Q adalah volume kendaraan, N adalah jumlah kendaraan terdeteksi, dan T adalah durasi pengamatan. Karena kendaraan memiliki karakteristik berbeda, jumlah kendaraan dikonversi ke SMP menggunakan Persamaan (2).

$$Q_{SMP} = \frac{\sum_{i=1}^{n} (N_i \times P_i)}{T} \tag{2}$$

Pada Persamaan (2), N_i adalah jumlah kendaraan kelas ke-i, P_i adalah faktor konversi SMP, dan T adalah durasi pengamatan. Faktor SMP yang digunakan adalah 1,0 untuk mobil, 0,4 untuk sepeda motor, dan 1,3 untuk

kendaraan berat. Selanjutnya, DS dihitung menggunakan Persamaan (3).

$$DS = \frac{Q_{SMP}}{C} \quad (3)$$

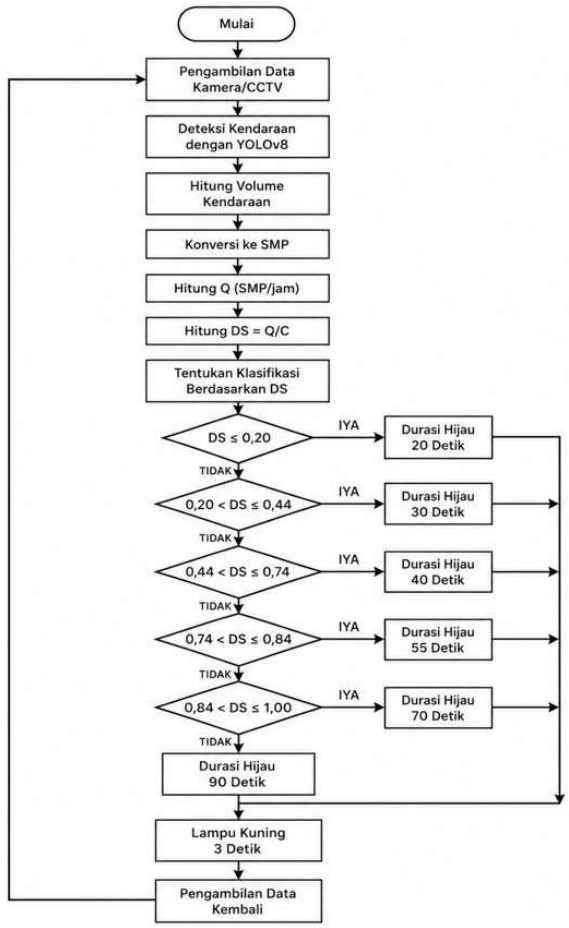
Pada Persamaan (3), C adalah kapasitas jalan dalam SMP/jam. Nilai DS digunakan untuk menentukan kategori volume lalu lintas dan durasi lampu hijau sebagaimana ditunjukkan pada Tabel 2.

TABEL 2
Klasifikasi DS dan durasi lampu hijau adaptif.

Kategori	Derajat Kejenuhan	Hijau
Sangat rendah	$DS \leq 0{,}20$	20 s
Rendah	$0{,}20 < DS \leq 0{,}44$	30 s
Sedang	$0{,}44 < DS \leq 0{,}74$	40 s
Tinggi	$0{,}74 < DS \leq 0{,}84$	55 s
Sangat tinggi	$0{,}84 < DS \leq 1{,}00$	70 s
Jenuh/macet	$DS > 1{,}00$	90 s

D. Logika Penentuan Durasi Hijau

Flowchart penentuan durasi hijau ditunjukkan pada Gambar 3. Alur dimulai dari pengambilan data CCTV, deteksi kendaraan menggunakan YOLOv8, konversi volume ke SMP, perhitungan Q SMP/jam dan DS, kemudian penentuan durasi hijau berdasarkan kategori DS. Setelah fase hijau selesai, sistem menambahkan lampu kuning selama 3 detik dan kembali melakukan pengamatan pada siklus berikutnya.



GAMBAR 3
Flowchart penentuan durasi lampu hijau adaptif berdasarkan nilai DS.

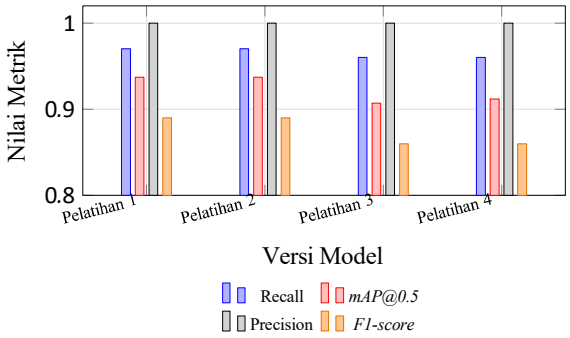
E. Implementasi Perangkat Keras dan Perangkat Lunak

Sistem diimplementasikan menggunakan Raspberry Pi 5 RAM 8 GB sebagai pusat pemrosesan. Perangkat lunak dibangun menggunakan Python, OpenCV, Ultralytics YOLOv8, Tkinter, NumPy, PIL/Pillow, serta *library* GPIO Raspberry Pi. Komponen keluaran terdiri atas empat modul *traffic light*, empat *display countdown* TM1637, breadboard, kabel jumper, dan miniatur persimpangan empat arah. Dashboard monitoring digunakan untuk menampilkan video empat arah, status lampu, arah aktif, arah yang diamati, data perhitungan, dan rekap kendaraan.

IV. HASIL DAN PEMBAHASAN

A. Performa Pelatihan Model YOLOv8

Performa model YOLOv8 berdasarkan empat tahapan pelatihan ditunjukkan pada Gambar 4. Seluruh versi model menghasilkan metrik di atas 0,80, sehingga secara umum memenuhi kriteria performa deteksi yang memadai. Precision bernilai 1,00 pada seluruh versi, sedangkan *recall* berada pada rentang 0,96–0,97. Nilai *mAP@0.5* tertinggi diperoleh pada Pelatihan 1 dan Pelatihan 2 sebesar 0,937, sedangkan Pelatihan 4 memperoleh *mAP@0.5* sebesar 0,912. Nilai *F1-score* Pelatihan 1 dan Pelatihan 2 sebesar 0,89, sedangkan Pelatihan 3 dan Pelatihan 4 sebesar 0,86.



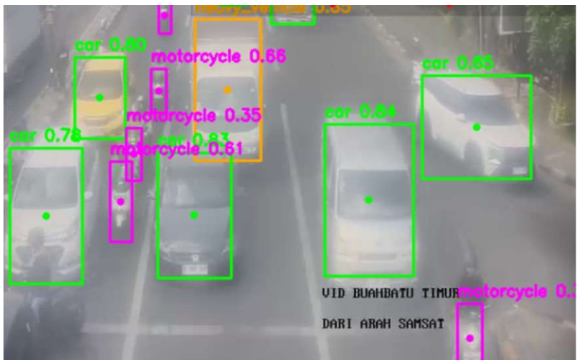
GAMBAR 4
Perbandingan performa model YOLOv8 berdasarkan tahapan pelatihan.

Meskipun Pelatihan 1 dan Pelatihan 2 memiliki nilai metrik numerik lebih tinggi, model Pelatihan 4 dipilih sebagai model final karena menggunakan dataset lebih besar dan menunjukkan generalisasi visual lebih baik pada pengujian *frame* CCTV siang dan malam. Pada kondisi siang, model Pelatihan 4 mampu mendeteksi mobil, sepeda motor, dan kendaraan berat secara lebih seimbang. Pada kondisi malam, Pelatihan 4 lebih baik dalam mengenali kendaraan berat yang sebelumnya dapat salah diklasifikasikan sebagai mobil oleh model lain. Pertimbangan ini penting karena sistem tidak hanya memerlukan skor metrik tinggi, tetapi juga kestabilan klasifikasi untuk menghasilkan perhitungan SMP yang lebih representatif.

B. Hasil Deteksi pada Kondisi Siang dan Malam

Hasil deteksi pada kondisi siang dan malam ditunjukkan pada Gambar 5 dan Gambar 6. Pada kondisi siang, pencahayaan yang lebih baik membuat bentuk kendaraan, batas objek, dan area jalan lebih mudah dikenali oleh model. Kendaraan dapat diberi *bounding box* dengan label kelas dan *confidence*. Hal ini menunjukkan bahwa model mampu digunakan sebagai

modul persepsi visual untuk menghitung kendaraan pada video CCTV.



GAMBAR 5
Hasil deteksi kendaraan menggunakan YOLOv8 pada kondisi siang hari.

Pada kondisi malam, sistem tetap mampu mendeteksi kendaraan, tetapi performanya lebih dipengaruhi oleh pencahayaan rendah, pantulan lampu kendaraan, *blur*, dan kualitas video CCTV. Objek kecil seperti sepeda motor cenderung lebih sulit dideteksi dibandingkan mobil. Oleh karena itu, variasi dataset siang dan malam menjadi faktor penting dalam menjaga kemampuan generalisasi model.



GAMBAR 6
Hasil deteksi kendaraan menggunakan YOLOv8 pada kondisi malam hari.

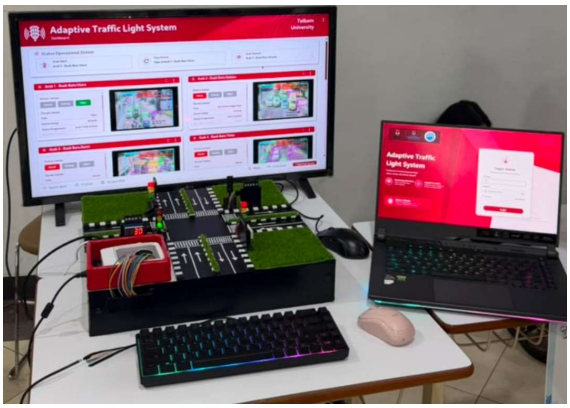
C. Dashboard Monitoring dan Prototipe

Dashboard monitoring ditunjukkan pada Gambar 7. Dashboard berfungsi sebagai antarmuka pemantauan yang menampilkan video CCTV, hasil deteksi, status lampu, arah aktif, arah pengamatan, data perhitungan volume, serta status sistem. Antarmuka ini memudahkan pengguna untuk memahami kondisi sistem secara *real-time*.



GAMBAR 7
Tampilan *dashboard monitoring* sistem lampu lalu lintas adaptif.

Implementasi prototipe fisik ditunjukkan pada Gambar 8. Raspberry Pi 5 mengendalikan modul *traffic light* dan *display countdown* TM1637 melalui GPIO. Lampu merah, kuning, dan hijau dapat menyala sesuai fase yang ditentukan sistem, sedangkan TM1637 menampilkan hitungan mundur durasi fase. Dengan demikian, hasil keputusan perangkat lunak dapat direpresentasikan dalam bentuk keluaran fisik pada miniatur persimpangan.



GAMBAR 8
Prototipe fisik sistem lampu lalu lintas adaptif berbasis Raspberry Pi.

D. Simulasi *Adaptive* dan *Fixed-Time*

Simulasi dilakukan menggunakan Python berdasarkan alur perhitungan dan logika pengaturan durasi hijau yang dijelaskan pada buku Tugas Akhir. Algoritma simulasi dibuat mengikuti mekanisme sistem yang dirancang, yaitu membaca volume kendaraan, mengonversi volume ke SMP, menghitung Q SMP/jam dan DS, kemudian menentukan durasi lampu hijau adaptif sesuai kategori kepadatan lalu lintas. Hasil sistem adaptif kemudian dibandingkan dengan sistem *fixed-time*, yaitu sistem lampu lalu lintas dengan durasi hijau tetap tanpa penyesuaian terhadap perubahan volume kendaraan. Lima skenario digunakan untuk merepresentasikan variasi kondisi lalu lintas, sedangkan parameter pembanding meliputi total kendaraan datang, kendaraan terlayani, persentase pelayanan, rata-rata antrean, dan rata-rata waktu tunggu.

TABEL 3
Parameter volume kendaraan pada setiap skenario simulasi.

Skenario dan Kondisi	Utara	Selatan	Timur	Barat
Skenario 1 – Volume tidak seimbang	1700	3200	800	500
Skenario 2 – Volume rendah merata	800	900	700	1000
Skenario 3 – Volume sedang merata	1800	2000	1600	1900
Skenario 4 – Volume tinggi tidak seimbang	3200	3600	2200	2800
Skenario 5 – Salah satu arah sangat padat	1200	800	2800	800

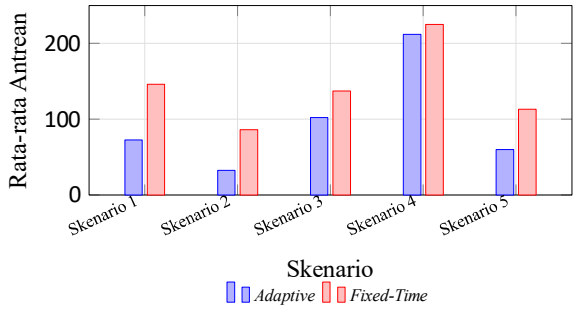
Keterangan: seluruh volume kendaraan dinyatakan dalam SMP/jam.

Parameter volume kendaraan pada setiap skenario simulasi ditunjukkan pada Tabel 3. Nilai kapasitas acuan (C) yang digunakan pada simulasi adalah 2800 SMP/jam untuk setiap arah.

TABEL 4
Rekap rata-rata simulasi *adaptive* dan *fixed-time*.

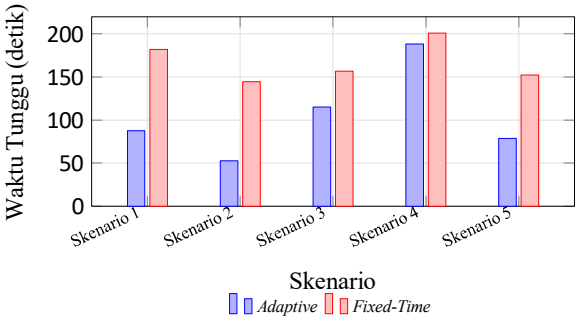
Parameter	*Adaptive*	*Fixed-Time*
Total kendaraan datang	3216,6	3216,6
Total kendaraan terlayani	3114,2	3049,4
Persentase terlayani	97,08%	94,92%
Rata-rata antrean	95,96	141,66
Rata-rata waktu tunggu	104,44 s	167,24 s

Berdasarkan Tabel 4, sistem adaptif melayani rata-rata 64,8 kendaraan lebih banyak dibandingkan *fixed-time*. Persentase kendaraan terlayani meningkat dari 94,92% menjadi 97,08%. Selain itu, rata-rata antrean berkurang dari 141,66 kendaraan menjadi 95,96 kendaraan, atau turun sekitar 32,26%. Rata-rata waktu tunggu juga turun dari 167,24 detik menjadi 104,44 detik, atau berkurang sekitar 37,55%.



GAMBAR 9
Perbandingan rata-rata antrean antara sistem adaptif dan *fixed-time*.

Berdasarkan Gambar 9, sistem adaptif menghasilkan antrean lebih rendah pada seluruh skenario. Penurunan antrean paling terlihat pada skenario dengan volume tidak seimbang dan salah satu arah sangat padat. Hal ini menunjukkan bahwa alokasi durasi hijau berdasarkan kategori DS mampu memberikan prioritas lebih baik pada arah dengan beban kendaraan lebih besar.



GAMBAR 10
Perbandingan rata-rata waktu tunggu antara sistem adaptif dan *fixed-time*.

Berdasarkan Gambar 10, waktu tunggu sistem adaptif lebih rendah dibandingkan *fixed-time* pada seluruh skenario. Hal ini terjadi karena *fixed-time* memberikan durasi hijau yang sama atau tetap tanpa mempertimbangkan variasi volume, sedangkan sistem adaptif menyesuaikan durasi hijau berdasarkan hasil pengamatan, konversi SMP, Q SMP/jam, dan DS. Dengan demikian, sistem adaptif lebih responsif terhadap perubahan kondisi lalu lintas.

E. Pembahasan Umum

Hasil penelitian menunjukkan bahwa sistem yang dirancang berhasil menghubungkan proses deteksi kendaraan dengan pengaturan durasi lampu hijau adaptif. Keunggulan utama sistem adalah integrasi antara *computer vision* dan parameter lalu lintas. Deteksi kendaraan tidak hanya digunakan untuk menampilkan *bounding box*, tetapi juga menjadi masukan perhitungan volume, konversi SMP, Q SMP/jam, DS, dan keputusan durasi hijau. Integrasi ini membuat sistem lebih dekat dengan kebutuhan pengaturan lalu lintas dibandingkan pendekatan yang hanya melakukan deteksi atau counting kendaraan.

Meskipun demikian, sistem masih memiliki keterbatasan. Sistem belum diterapkan langsung pada APILL publik dan masih berada pada skala prototipe. Kinerja deteksi masih dipengaruhi kualitas video CCTV, kondisi pencahayaan, sudut kamera, kepadatan kendaraan, serta kemungkinan kendaraan saling menutupi. Selain itu, pemrosesan *multi-stream* secara *real-time* pada Raspberry Pi masih membutuhkan optimasi lebih lanjut agar stabil pada durasi operasi panjang. Keputusan durasi hijau juga masih berbasis volume kendaraan dan DS, belum memasukkan parameter lain seperti panjang antrean aktual, pejalan kaki, kendaraan prioritas, atau koordinasi antar simpang.

V. KESIMPULAN

Penelitian ini berhasil merancang prototipe sistem pengaturan lampu lalu lintas adaptif berbasis deteksi volume kendaraan menggunakan YOLOv8 dan Raspberry Pi 5. Sistem mampu memproses video CCTV, mendeteksi tiga kelas kendaraan, melakukan *vehicle counting*, mengonversi hasil deteksi ke satuan SMP, menghitung Q SMP/jam dan DS, serta menentukan durasi lampu hijau adaptif berdasarkan kategori volume lalu lintas. Sistem juga berhasil diintegrasikan dengan *dashboard monitoring*, modul *traffic light*, dan *display countdown* TM1637 pada prototipe fisik.

Hasil pelatihan model menunjukkan bahwa seluruh versi YOLOv8 memiliki performa deteksi baik dengan nilai metrik di atas 0,80. Model Pelatihan 4 dipilih sebagai model final karena memiliki keseimbangan yang lebih sesuai antara generalisasi visual dan kebutuhan klasifikasi kendaraan pada kondisi CCTV siang dan malam. Hasil simulasi menunjukkan bahwa sistem adaptif lebih unggul dibandingkan *fixed-time*, dengan rata-rata kendaraan terlayani sebesar 3114,2 kendaraan, rata-rata antrean 95,96 kendaraan, dan rata-rata waktu tunggu 104,44 detik. Dibandingkan *fixed-time*, sistem adaptif menurunkan antrean sekitar 32,26% dan waktu tunggu sekitar 37,55%.

Pengembangan selanjutnya perlu difokuskan pada perluasan dataset, peningkatan akurasi counting dengan metode *tracking*, optimasi inferensi pada Raspberry Pi, serta pengujian pada lebih banyak lokasi dan durasi pengamatan yang lebih panjang. Sistem juga dapat dikembangkan dengan menambahkan parameter lain seperti panjang antrean, waktu tunggu aktual, kendaraan prioritas, pejalan kaki, dan koordinasi antar simpang.

REFERENSI

[1] S. O. Subair, B. A. Ibitoye, and A. T. Kuranga, "Evaluation of traffic congestion in an urban roads: A review," *ABUAD Journal of Engineering and Applied Sciences*, vol. 2, no. 2, pp. 1–7, Aug. 2024.

[2] I. K. Hachim and H. L. Swadi, "Multi-objective optimization of traffic signal timings for minimizing waiting time, co2 emissions, and fuel consumption at intersections," *Kufa Journal of Engineering*, vol. 15, no. 3, pp. 21–31, Jul. 2024.

[3] S. Rao, "Signal timing optimization in urban and suburban networks," *ESP Journal of Engineering & Technology Advancements*, vol. 5, no. 2, pp. 155–162, May 2025.

[4] Majstorovic´, L. Tišljaric´, E. Ivanjko, and T. Caric´, "Urban traffic signal control under mixed traffic flows: Literature review," *Applied Sciences*, vol. 13, no. 7, p. 4484, Apr. 2023.

[5] M. A. Ottom and A. Al-Omari, "An adaptive traffic lights system using machine learning," *The International Arab Journal of Information Technology*, vol. 20, no. 3, pp. 407–418, May 2023.

[6] A. Agrahari, M. M. Dhabu, P. S. Deshpande, A. Tiwari, M. A. Baig, and A. D. Sawarkar, "Artificial intelligence-based adaptive traffic signal control system: A comprehensive review," *Electronics*, vol. 13, no. 19, p. 3875, Sep. 2024.

[7] K. S. Mehta, K. N. Raj, and K. N. Brahmbhatt, "Machine learning solutions for adaptive traffic signal control: A review of image-based approaches," *World Journal of Advanced Engineering Technology and Sciences*, vol. 13, no. 1, pp. 476–481, 2024.

[8] I. G. Erdagi, S. Gavric, and A. Stevanovic, "Impact of adaptive traffic control system implementation methods on operations of a coordinated network," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2679, no. 6, pp. 963–992, 2025.

[9] W. Karoon, P. Chuasuai, P. Thipprasert, N. Khongchu, P. Kunakornjittirak, and T. Siriborvornratanakul, "Adaptive traffic light control using vision-based deep learning for vehicle density estimation," in *Proc. 2024 6th Asia Pacific Information Technology Conference (APIT)*, Bangkok, Thailand, Jan. 2024, pp. 37–42.

[10] Z. Jing, S. Li, and Q. Zhang, "Yolov8-ste: Enhancing object detection performance under adverse weather conditions with deep learning," *Electronics*, vol. 13, no. 24, p. 5049, Dec. 2024.

[11] Y. Cheng and B. Liu, "An improved yolov8-based vehicle detection algorithm for complex traffic scenes," *Scientific Journal of Intelligent Systems Research*, vol. 7, no. 11, pp. 15–26, 2025.

[12] D. J. Marcelleno and M. P. K. Putra, "Performance evaluation of yolov8 in real-time vehicle detection in various environmental conditions," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 1, pp. 269–279, Feb. 2025.

[13] W. P. N. Putra, A. I. Pradana, and Nurchim, "Implementasi sistem penghitungan volume kendaraan menggunakan yolov8," *Jurnal FASILKOM (Teknologi Informasi dan Ilmu Komputer)*, vol. 14, no. 2, pp. 443–450, Aug. 2024.

[14] Nadiyah, B. Yusman, M. Furqan, and Z. Arifin, "Sistem analisis kepadatan lalu lintas menggunakan teknologi yolov8," *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, vol. 6, no. 1, pp. 90–97, 2025.

[15] Republik Indonesia, "Undang-undang republik indonesia nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan," 2009, lembaran Negara RI Tahun 2009 No. 96, Tambahan Lembaran Negara RI No. 5025, Jakarta.

[16] Menteri Perhubungan Republik Indonesia, "Peraturan menteri perhubungan republik indonesia nomor pm 49 tahun 2014 tentang alat pemberi isyarat lalu lintas," 2014, berita Negara RI, Jakarta.

[17] Direktorat Jenderal Perhubungan Darat, "Peraturan direktur jenderal perhubungan darat tentang petunjuk teknis alat pemberi isyarat lalu lintas," 2021, kementerian Perhubungan Republik Indonesia, Jakarta, Indonesia.

[18] N. Bilous, V. Malko, M. Frohme, and A. Nechyporenko, "Comparison of cnn-based architectures for detection of different object classes," 2024, preprints, preprints202409.1310.v1, doi: 10.20944/preprints202409.1310.v1.

[19] S. Y. Mohammed, "Architecture review: Two-stage and one-stage object detection," *Franklin Open*, vol. 12, p. 100322, 2025.

[20] Kementerian Pekerjaan Umum dan Perumahan Rakyat, Direktorat Jenderal Bina Marga, "Pedoman kapasitas jalan indonesia," 2023, surat Edaran Nomor: 21/SE/Db/2023, Jakarta, Indonesia.

[21] Direktorat Jenderal Bina Marga, "Manual kapasitas jalan indonesia (mkji) 1997," 1997, departemen Pekerjaan Umum, Jakarta, Indonesia.