

Analisis Komputasi dan Waktu Inferensi Model YOLOv8 dan YOLOv11 pada Deteksi Kendaraan

^{1*}Irwan April Liansah, ²Danar Putra Pamungkas, ³Made Ayu Dusea
Widyadara

^{1,2,3} Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹irwancious@gmail.com, ²danar@unpkediri.ac.id, ³madedara@unpkediri.ac.id

Penulis Korespondens : Irwan April Liansah

Abstrak—Pemantauan lalu lintas cerdas memerlukan model deteksi objek yang akurat dan efisien secara komputasi untuk diimplementasikan pada perangkat tepi. Penelitian ini bertujuan membandingkan tolak tarik performa antara arsitektur YOLOv8 dan YOLOv11 berdasarkan stabilitas pelatihan, ukuran model, dan kecepatan inferensi. Pendekatan eksperimental kuantitatif digunakan dengan melatih kedua model selama 50 iterasi menggunakan dataset citra lalu lintas pada lingkungan komputasi awan. Hasil pengujian menunjukkan YOLOv11 lebih efisien dalam ruang penyimpanan (5,22 Megabyte) dan unggul pada akurasi keseluruhan dengan nilai 77,8 persen, khususnya pada deteksi kendaraan besar. Sebaliknya, YOLOv8 menunjukkan konvergensi pelatihan yang lebih stabil di fase awal dan kecepatan inferensi yang lebih unggul (7,31 milidetik per bingkai) dibandingkan YOLOv11 (9,25 milidetik). Disimpulkan bahwa YOLOv8 direkomendasikan untuk implementasi waktu nyata guna mencegah penundaan pemrosesan, sedangkan YOLOv11 ideal untuk perangkat dengan kapasitas memori terbatas yang menuntut presisi tinggi.

Kata Kunci— Deteksi Kendaraan, Kecepatan Inferensi, Komputasi, YOLOv11, YOLOv8

Abstract— Intelligent traffic monitoring requires computationally efficient object detection models for edge device implementation. This study compares the performance trade-off between YOLOv8 and YOLOv11 architectures based on training stability, model size, and inference speed. A quantitative experimental approach was used by training both models for 50 iterations using a traffic image dataset in a cloud computing environment. The results show YOLOv11 is more storage-efficient (5.22 Megabytes) and excels in overall accuracy (77.8 percent), particularly in detecting large vehicles. Conversely, YOLOv8 demonstrates more stable training convergence in the initial phase and superior inference speed (7.31 milliseconds per frame) compared to YOLOv11 (9.25 milliseconds). It is concluded that YOLOv8 is recommended for real-time implementation to prevent processing delays, while YOLOv11 is ideal for devices with limited memory capacity demanding high precision.

Keywords— Vehicle Detection, Inference Speed, Computation, YOLOv11, YOLOv8

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Sistem Transportasi Cerdas (*Intelligent Transportation Systems/ITS*) memiliki peran esensial dalam mengelola dan memantau volume kendaraan secara otomatis guna mengurangi kemacetan perkotaan [1]. Tantangan utama dalam implementasi ITS berbasis visi komputer di lapangan adalah keterbatasan sumber daya komputasi pada perangkat tepi (*edge devices*), yang menuntut penggunaan model pendeteksi dengan keseimbangan optimal antara efisiensi komputasi dan

akurasi. Model yang menuntut beban komputasi berat akan memicu penundaan pemrosesan atau *frame drop*, sehingga mengurangi keandalan sistem pelacakan secara *real-time* [2].

Berbagai penelitian terdahulu telah mengadopsi arsitektur algoritma *You Only Look Once* (YOLO) berkat kemampuannya memecahkan masalah keseimbangan tersebut. Varian model ringan seperti YOLOv8n yang dikombinasikan dengan DeepSORT mampu mencapai rata-rata kecepatan 19,4 *Frame Per Second* (FPS) ketika dijalankan pada perangkat Jetson Nano [3]. Studi serupa mengonfirmasi keandalan YOLOv8 yang dipadukan dengan *Centroid Tracking* dalam mempertahankan identitas objek kendaraan serta mencegah penghitungan ganda pada modul berspesifikasi rendah [4]. Pada konteks kepadatan jalan, YOLOv8 juga terbukti mampu mengklasifikasikan situasi kemacetan secara langsung (*live video streams*) untuk membantu otomatisasi manajemen lampu lalu lintas [5].

Di sisi lain, pemanfaatan model arsitektur iterasi terbaru, YOLOv11, mulai dieksplorasi dan diklaim sebagai pendeteksi kendaraan yang sangat ringan (ZZ-YOLOv11) untuk menghadapi kompleksitas kondisi pencahayaan di lingkungan urban [6]. Secara struktural, YOLOv11 memperkenalkan mekanisme fitur spasial baru yang mampu menekan dan mereduksi total parameter hingga 22% lebih kecil dibandingkan varian dari YOLOv8, sekaligus mempertahankan metrik rata-rata presisi (*Mean Average Precision* / mAP) di tingkat yang tinggi [7].

Meskipun YOLOv11 menawarkan pembaruan pada ekstraksi fitur dan penekanan jumlah parameter, komparasi langsung secara empiris antara YOLOv8 dan YOLOv11 terhadap metrik beban komputasi di lingkungan perangkat keras yang seragam masih belum banyak dipublikasikan [6], [7]. Mayoritas publikasi terdahulu hanya berfokus pada hasil peningkatan metrik akurasi deteksi, tanpa melakukan analisis spesifik mengenai fenomena tolak tarik (*trade-off*) antara kecepatan inferensi dan ruang penyimpanan yang krusial bagi arsitektur komputasi perangkat pinggir.

Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi dan melakukan analisis komparatif performa komputasi antara model YOLOv8 dan YOLOv11 pada kasus deteksi lalu lintas. Secara spesifik, penelitian ini membedah perbedaan ukuran model penyimpanan, kecepatan waktu inferensi per *frame*, serta stabilitas kurva konvergensi pada proses iterasi pelatihan. Temuan dari penelitian ini diharapkan dapat memberikan rekomendasi teknis yang jelas mengenai arsitektur mana yang paling optimal untuk diimplementasikan ke dalam modul sistem pemantauan lalu lintas mandiri.

II. METODE

Penelitian ini menggunakan pendekatan eksperimental kuantitatif. Materi yang diteliti adalah komparasi dua arsitektur *deep learning* dari keluarga YOLO, yaitu YOLOv8 dan YOLOv11, yang telah terbukti efisien dalam pemrosesan visual waktu nyata [8]. Tahapan pelaksanaan penelitian dirancang secara terstruktur mulai dari persiapan hingga evaluasi akhir. Secara visual, alur penelitian komparatif ini dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alur Penelitian Komparasi Model

Untuk memastikan kejelasan dan keterulangan (*repeatability*) penelitian, tahapan pada Gambar 1 dijabarkan ke dalam tiga skenario utama berikut:

A. Persiapan Dataset dan Lingkungan Komputasi

Tahap awal meliputi penyiapan *dataset* citra lalu lintas dan penentuan perangkat keras. Untuk menghindari bias spesifikasi (*hardware bias*) selama proses komparasi beban komputasi, seluruh skenario pengujian dilakukan pada lingkungan *cloud* yang terstandarisasi [8]. Rincian spesifikasi perangkat keras dan lunak yang digunakan dijabarkan pada Tabel 1.

Tabel 1. Spesifikasi Lingkungan Komputasi

Komponen	Spesifikasi
Lingkungan (Environment)	Google Colaboratory
Graphics Processing Unit (GPU)	NVIDIA Tesla T4 15GB
RAM Sistem	12,7 GB
Bahasa Pemrograman	Python 3.12
Framework Deep Learning	Ultralytics 8.4.46

B. Konfigurasi Pelatihan Model

Tahap kedua berfokus pada proses pelatihan (*training*) kedua model menggunakan *dataset* yang telah disiapkan. Pelatihan dikonfigurasi berjalan selama 50 *epoch* dengan ukuran citra (*image size*) 640 piksel. Variabel yang diukur pada tahap ini adalah perilaku konvergensi dan tingkat akurasi akhir model. Pengukuran dievaluasi menggunakan tiga metrik utama: (1) metrik *box_loss* untuk mengukur presisi koordinat kotak pembatas, (2) metrik *dfl_loss* (*Distribution Focal Loss*) untuk mengevaluasi ketajaman tepi lokalisasi pada objek yang saling berdempetan atau oklusi, dan (3) mAP50 (*Mean Average Precision*) sebagai standar validasi akurasi deteksi objek secara keseluruhan [9]. Data diekstraksi dari *log* visual dan numerik yang dihasilkan sistem secara otomatis selama proses iterasi.

C. Pengujian Inferensi

Tahap ketiga adalah pengujian implementasi menggunakan file pembobotan terbaik (*best.pt*) yang dihasilkan dari proses pelatihan. Model YOLOv8 dan YOLOv11 diuji menggunakan masukan (*input*) citra lalu lintas yang sama secara bergantian. Pada tahap ini, pengujian difokuskan pada pengukuran kecepatan model berdasarkan total waktu pemrosesan per bingkai gambar (*inference time*) [10]. Kecepatan ini dihitung dengan mengakumulasi durasi pra-pemrosesan (*preprocessing*), durasi inferensi jaringan saraf (*network inference*), dan pasca-pemrosesan (*post-processing*). Secara matematis, total waktu inferensi dirumuskan pada Persamaan 1.

$$T_{Total} = t_{pre} + t_{inf} + t_{post} \quad (1)$$

Di mana T_{Total} adalah total waktu pemrosesan, t_{pre} adalah waktu penyiapan citra, t_{inf} adalah waktu prediksi oleh model, dan t_{post} adalah waktu penyaringan kotak pembatas (*Non-Maximum Suppression*).

D. Komparasi Hasil Komputasi

Tahap terakhir adalah melakukan komparasi atau perbandingan analitis dari data yang diperoleh pada tahap pengujian. Variabel metrik yang dikomparasikan meliputi dua parameter

utama, yaitu: (1) Ukuran ruang penyimpanan model dalam satuan Megabyte (MB), dan (2) Kecepatan total waktu inferensi dalam satuan milidetik (ms). Evaluasi dan komparasi pada kedua aspek ini sangat esensial untuk menentukan model mana yang memiliki tolak tarik (*trade-off*) paling optimal. Hal ini dikarenakan implementasi sistem pemantauan di lapangan umumnya menggunakan perangkat tepi (*edge devices*) yang memiliki keterbatasan memori fisik penyangga dan daya komputasi prosesor [11].

III. HASIL DAN PEMBAHASAN

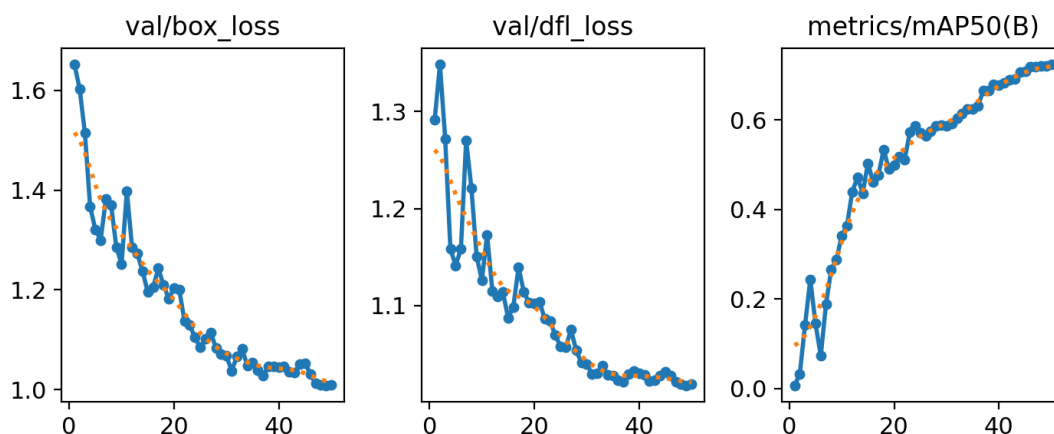
Pembahasan Pembahasan terhadap hasil penelitian dan pengujian yang diperoleh disajikan dalam bentuk uraian teoritik secara kualitatif maupun kuantitatif. Hasil dievaluasi secara berurutan sesuai dengan tahapan skenario yang telah dirancang.

A. Hasil Persiapan Dataset dan Lingkungan Komputasi

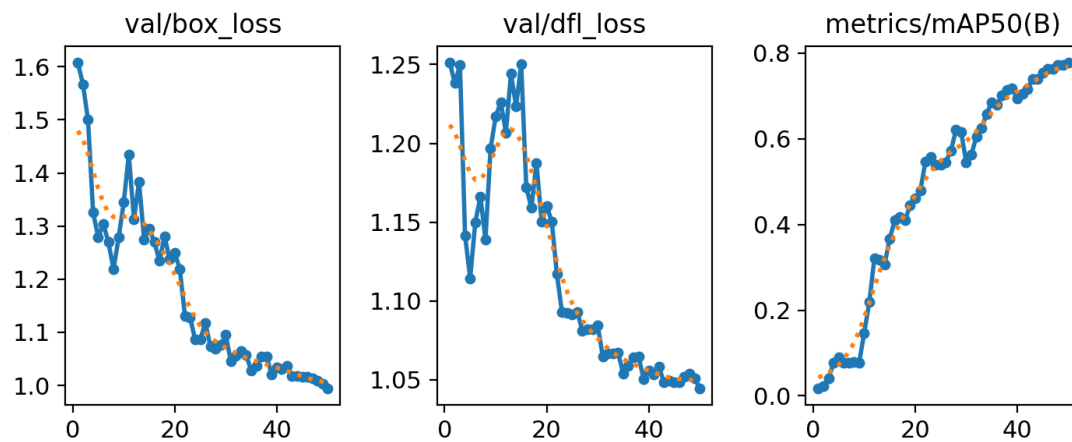
Tahap awal pemrosesan data telah berhasil menghimpun sampel citra lalu lintas yang representatif untuk kebutuhan pelatihan. *Dataset* yang digunakan bersumber dari ekstraksi rekaman kondisi nyata pada dua lokasi strategis, yaitu Simpang Nabatiyasa dan kawasan persimpangan Semampir. Penggabungan dua titik lokasi ini memberikan keuntungan secara visual guna memastikan kelengkapan dan variasi kelas kendaraan yang terdeteksi, khususnya untuk mengakomodasi tangkapan objek kendaraan besar seperti bus yang lebih masif melintas di area Semampir. Seluruh data tersebut kemudian diunggah dan dieksekusi pada lingkungan Google Colab menggunakan GPU T4 yang terbukti stabil menjalankan proses tanpa kendala kehabisan memori (*Out of Memory*).

B. Evaluasi Konvergensi Pelatihan

Berdasarkan skenario pelatihan model selama 50 *epoch*, evaluasi komparatif dilakukan dengan menganalisis metrik kurva pembelajaran dan hasil validasi akhir dari kedua arsitektur. Fokus evaluasi ditekankan pada stabilitas penurunan tingkat kesalahan (*loss*) serta pencapaian metrik rata-rata presisi (*Mean Average Precision / mAP50*).



Gambar 2. Kurva Metrik Pelatihan YOLOv8



Gambar 3. Kurva Metrik Pelatihan YOLOv11

Berdasarkan perbandingan visual antara Gambar 2 dan Gambar 3 yang divalidasi oleh log sistem pelatihan, terlihat perbedaan perilaku konvergensi yang signifikan pada rentang iterasi awal. Pada arsitektur YOLOv8 (Gambar 2), model menunjukkan stabilitas pembelajaran yang sangat konsisten. Hal ini dibuktikan dengan tren penurunan yang mulus pada kurva `val/box_loss` maupun `val/dfl_loss`. Stabilitas ini menandakan bahwa YOLOv8 mampu beradaptasi dengan pola *dataset* kendaraan tanpa mengalami kesulitan berarti di fase awal penyesuaian bobot.

Sebaliknya, arsitektur YOLOv11 (Gambar 3) memperlihatkan pola pembelajaran yang sangat fluktuatif. Terdapat lonjakan kesalahan (*spike*) yang cukup ekstrem di sekitar *epoch* ke-10 hingga ke-15. Fluktuasi ini mengindikasikan bahwa mekanisme ekstraksi fitur pada YOLOv11 mengalami fase *gradient instability* (ketidakstabilan adaptasi) saat pertama kali mengenali variasi objek kendaraan.

Meskipun YOLOv11 menunjukkan ketidakstabilan di fase awal, arsitektur ini mencatatkan performa pemulihan (*recovery*) yang masif dan menghasilkan akurasi akhir yang lebih superior. Berdasarkan log validasi pada *epoch* ke-50, YOLOv11 berhasil mencapai tingkat akurasi keseluruhan (mAP50) sebesar 0.778 (77,8%). Angka ini mengungguli YOLOv8 yang tertahan dan mengalami pelambatan peningkatan (*plateau*) di angka 0.725 (72,5%).

Lebih lanjut, pembedahan metrik akurasi berdasarkan kelas (*class-specific validation*) mengungkap karakteristik ekstraksi fitur dari masing-masing model. Arsitektur YOLOv11 menunjukkan keunggulan yang sangat absolut dalam mendeteksi objek kendaraan berukuran besar. Pada kelas "bus", YOLOv11 mencatatkan mAP50 sebesar 0.920, jauh meninggalkan YOLOv8 yang hanya mampu mencapai 0.668. Hal ini membuktikan bahwa pembaruan mekanisme *spatial feature* pada YOLOv11 sangat efektif untuk menangkap objek berdimensi masif. Di sisi lain, YOLOv8 masih mempertahankan sedikit keunggulan pada lokalisasi kendaraan berukuran sedang dan kecil, di mana model ini mencatatkan mAP50 sebesar 0.817 untuk kelas "mobil" dan 0.713 untuk kelas "motor", berbanding tipis dengan YOLOv11 yang mencatatkan angka 0.789 dan 0.695 untuk kelas yang sama. Rincian komparasi metrik validasi akhir dari kedua model untuk seluruh kelas kendaraan dapat dilihat secara lengkap pada Tabel 2.

Tabel 2. Komparasi Metrik Validasi Akhir

Kelas Kendaraan	Metrik Evaluasi	YOLOv8	YOLOv11
Keseluruhan (<i>All</i>)	mAP50	0,725	0,778
	Presisi (<i>Precision</i>)	0,650	0,770
	<i>Recall</i>	0,686	0,728
Bus	mAP50	0,668	0,920
Mobil	mAP50	0,817	0,789
Motor	mAP50	0,713	0,695
Truk	mAP50	0,700	0,709

C. Hasil Pengujian Inferensi

Pengujian waktu inferensi telah dieksekusi menggunakan file pembobotan terbaik (*best.pt*) dari kedua model. Berdasarkan pengujian terhadap sampel citra lalu lintas berukuran 640 piksel, sistem mengkalkulasi komponen waktu pemrosesan yang meliputi pra-pemrosesan, inferensi jaringan, dan pasca-pemrosesan. Hasil pendataan sistem menunjukkan bahwa arsitektur YOLOv8 mencatatkan total waktu inferensi sebesar 7,31 milidetik per bingkai citra. Di sisi lain, eksekusi komputasi pada model YOLOv11 dengan kondisi parameter masukan yang identik menghasilkan total waktu inferensi yang lebih lama, yakni sebesar 9,25 milidetik per bingkai citra.

D. Komparasi dan Analisis Komputasi

Data kuantitatif hasil pengujian inferensi beserta ekstraksi ukuran ruang penyimpanan fisik dari kedua model komputasi selanjutnya dikomparasikan. Komparasi ini bertujuan untuk melihat tolak tarik (*trade-off*) performa dari masing-masing arsitektur. Ringkasan hasil komparasi disajikan pada Tabel 3.

Tabel 3. Komparasi Beban Komputasi dan Kecepatan Inferensi

Model Arsitektur	Ukuran Model (MB)	Waktu Inferensi (ms)
YOLOv8	5,96	7,31
YOLOv11	5,22	9,25

Berdasarkan Tabel 3, terlihat fenomena tolak tarik spesifikasi yang sangat tegas. Dari segi efisiensi memori fisik, YOLOv11 terbukti lebih ringan dengan ukuran 5,22 MB, berbanding terbalik dengan YOLOv8 yang memakan ruang sebesar 5,96 MB. Pengurangan ukuran model ini memberikan implikasi struktural yang sangat menguntungkan apabila sistem diimplementasikan secara tertanam pada perangkat tepi (*edge devices*), seperti unit kamera berspesifikasi rendah.

Namun demikian, keunggulan YOLOv11 pada efisiensi ruang komputasi harus dibayar dengan penurunan kecepatan pemrosesan. YOLOv8 terbukti secara empiris lebih responsif dengan selisih kecepatan sekitar 1,94 milidetik lebih cepat dibandingkan YOLOv11. Pada ranah implementasi deteksi lalu lintas *real-time*, kecepatan eksekusi merupakan parameter krusial untuk mencegah penumpukan memori penyangga (*buffer*) yang dapat menyebabkan *frame drop*.

Batasan penelitian saat ini adalah komparasi komputasi baru dilakukan pada lingkungan komputasi awan. Untuk penelitian di masa depan, permasalahan yang belum terpecahkan terkait konsumsi daya (*power consumption*) dan suhu operasional (*thermal throttling*) saat model

dieksekusi secara terus-menerus pada perangkat mikrokontroler di luar ruangan perlu dikaji secara lebih komprehensif.

IV. KESIMPULAN

Berdasarkan hasil pengujian dan analisis komparatif yang telah dieksekusi, dapat disimpulkan bahwa pemilihan arsitektur model pendeteksi kendaraan sangat bergantung pada spesifikasi perangkat keras dan urgensi implementasi di lapangan. Arsitektur YOLOv11 memberikan kontribusi pembaruan yang signifikan dalam efisiensi ruang penyimpanan memori fisik dengan ukuran 5,22 MB, serta keunggulan tingkat akurasi deteksi puncak (mAP50) yang mencapai 77,8%, khususnya pada keandalan pengenalan objek berdimensi masif seperti bus. Namun demikian, untuk kebutuhan implementasi sistem transportasi cerdas berspesifikasi waktu nyata (*real-time*) yang menuntut respons seketika, arsitektur YOLOv8 secara empiris terbukti masih menjadi pilihan yang lebih tangguh. YOLOv8 menawarkan stabilitas konvergensi yang jauh lebih konsisten sejak awal fase iterasi pelatihan serta kecepatan waktu inferensi yang lebih singkat, yakni 7,31 milidetik per bingkai citra, sehingga sangat esensial untuk menekan risiko penumpukan antrean pemrosesan (*frame drop*). Sebagai saran untuk pengembangan ilmu pengetahuan di ranah visi komputer selanjutnya, penelitian di masa depan diharapkan dapat menguji keseimbangan kedua arsitektur ini secara langsung pada mikrokontroler perangkat tepi (*edge devices*) di luar ruangan guna mengukur ketahanan suhu (*thermal throttling*) dan efisiensi konsumsi daya operasionalnya.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika dan seluruh pihak terkait yang telah memberikan dukungan fasilitas komputasi maupun bimbingan akademik, sehingga penelitian mengenai analisis komparasi performa model pendeteksi kendaraan ini dapat terselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] J. Jusnidar, R. Riskayanti, Y. Setiawan, K. Khomsiah, A. Diath Yogmar, dan D. Tribuana, "IMPLEMENTASI WEB-BASED TRANSPORTATION MANAGEMENT SYSTEM DI INDUSTRI TRANSPORTASI: SEBUAH TINJAUAN SISTEMATIS LITERATUR," *Jurnal Teknologi dan Bisnis Cerdas*, vol. 1, no. 3, hlm. 194–209, Des 2025, doi: 10.64476/jtbc.v1i3.15.
- [2] N. H. Setiyawan, A. Hariyadi, dan A. Kurniawan, "Sistem Pengawasan CCTV Pada ATM Secara Real-Time Berbasis Internet of Things," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 5, no. 1, Mei 2025, doi: 10.59395/k0e60m33.
- [3] A. Saadeldin, M. M. Rashid, A. A. Shafie, dan T. F. Hasan, "Real-time vehicle counting using custom YOLOv8n and DeepSORT for resource-limited edge devices," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 22, no. 1, hlm. 104–112, Feb 2024, doi: 10.12928/TELKOMNIKA.v22i1.25096.
- [4] K. D. Dharmasaputra dan B. P. Hartato, "Evaluation of YOLOv8 and Centroid Tracking in Vehicle Detection, Classification, and Counting System," *Journal of Applied Informatics and Computing*, vol. 9, no. 5, hlm. 2310–2319, Okt 2025, doi: 10.30871/jaic.v9i5.10873.

- [5] Ms. Abirami S, Ajai B, Imran Hussain S, Kavin Ezhil Arasu S.U, dan Mohamed Yasinkhan R, “Real-Time Traffic Density Estimation Using Yolov8,” *International Research Journal on Advanced Engineering and Management (IRJAEM)*, vol. 4, no. 05, hlm. 1962–1967, Mei 2026, doi: 10.47392/IRJAEM.2026.0287.
- [6] Z. Zhang, Z. Zhang, G. Li, dan C. Xia, “ZZ-YOLOv11: A Lightweight Vehicle Detection Model Based on Improved YOLOv11,” *Sensors*, vol. 25, no. 11, 2025, doi: 10.3390/s25113399.
- [7] “YOLOv8 vs YOLO11: A Comprehensive Technical Comparison of Real-Time Vision Models | Ultralytics Docs.” Diakses: 5 Juni 2026. [Daring]. Tersedia pada: <https://docs.ultralytics.com/compare/yolov8-vs-yolo11#architectural-and-methodological-differences>
- [8] J. Terven, D.-M. Córdova-Esparza, dan J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, hlm. 1680–1716, 2023, doi: 10.3390/make5040083.
- [9] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, dan T. Huang, “UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios,” *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167190.
- [10] M. Sobirin dan P. Wijonarko, “Pemantauan Keamanan Real-Time pada Base Transceiver Station (BTS) Menggunakan YOLOv8 dan Integrasi Telegram: Optimasi Model dan Evaluasi Performa,” *ILKOMNIKA: Journal of Computer Science and Applied Informatics E*, vol. 7, no. 3, hlm. 533–547, 2025, doi: 10.28926/ilkomnika.v7i3.824.
- [11] H. Sukron dan S. Arifin, “Tinjauan Literatur tentang Perkembangan Wireless Sensor Network dalam Berbagai Bidang Aplikasi,” *Karapan Network Journal*, vol. 1, no. 01, 2025, doi: 10.20473/KNJ.01.01.pp-pp.