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

# IMPLEMENTASI YOLOv8n DAN DEEPSORT UNTUK DETEKSI KECELAKAAN SERTA ANALISIS LALU LINTAS PADA VIDEO CCTV JALAN RAYA

**Anak Agung Gede Dwitya Nugraha Parardhya✉, I Wayan Candra Winetra, I Nyoman Rai Widartha Kesuma**
Jurusan Teknologi Informasi, Politeknik Negeri Bali, Badung, Indonesia
Email: anakagunggede04@gmail.com

DOI: https://doi.org/10.46880/jmika.Vol10No2.pp1031-1045

***ABSTRACT***

*The increasing number of motorized vehicles has created challenges in traffic management, particularly in monitoring highway incidents. Conventional Closed-circuit Television (CCTV) surveillance relies heavily on human observation, making it prone to human error and delayed emergency responses. Therefore, this study proposes a web-based traffic analysis system utilizing Computer Vision technology. The YOLOv8n deep learning model is employed to perform accident detection, speed estimation, and vehicle counting simultaneously. The model was trained using 36 hyperparameter combinations in Google Colab. The best model achieved a mean Average Precision (mAP@50) of 0.862, with an mAP@50 of 0.881 for the accident class. Furthermore, the developed system was integrated with live CCTV streams and achieved an average processing speed of 25 Frames Per Second (FPS). The proposed system provides an integrated solution for real-time traffic analysis and accident detection, supporting more effective traffic monitoring.*

***Keyword: YOLOv8n, Computer Vision, Traffic Analysis, Accident Detection, DeepSORT.***

**ABSTRAK**

*Peningkatan jumlah kendaraan bermotor menimbulkan tantangan dalam pemantauan lalu lintas, khususnya dalam mendeteksi insiden di jalan raya. Pengawasan closed-circuit television (CCTV) konvensional yang bergantung pada observasi manusia rentan terhadap human error dan keterlambatan respons darurat. Oleh karena itu, penelitian ini mengusulkan sistem analisis lalu lintas berbasis web yang memanfaatkan teknologi Computer Vision. Model deep learning YOLOv8n diterapkan untuk melakukan deteksi kecelakaan, estimasi kecepatan, dan penghitungan jumlah kendaraan secara bersamaan. Pelatihan model dilakukan menggunakan 36 kombinasi hyperparameter di Google Colab. Hasil evaluasi menunjukkan bahwa model terbaik memperoleh nilai mean Average Precision (mAP@50) sebesar 0,862, dengan mAP@50 sebesar 0,881 pada kelas kecelakaan. Selanjutnya, sistem yang dikembangkan diintegrasikan dengan streaming CCTV secara langsung dan mampu memproses video dengan kecepatan rata-rata 25 Frames Per Second (FPS). Sistem yang dikembangkan memberikan solusi terpadu untuk analisis lalu lintas dan deteksi kecelakaan secara real-time sehingga dapat mendukung proses pemantauan lalu lintas yang lebih efektif.*

***Kata Kunci: YOLOv8n, Computer Vision, Analisis Lalu Lintas, Deteksi Kecelakaan, DeepSORT.***

## PENDAHULUAN

Transportasi jalan raya merupakan sektor strategis yang menjadi tulang punggung mobilitas masyarakat serta penggerak utama pertumbuhan ekonomi nasional. Seiring dengan pesatnya urbanisasi dan aktivitas ekonomi, volume kendaraan bermotor di Indonesia mengalami peningkatan yang sangat signifikan dari tahun ke tahun. Berdasarkan data Badan Pusat Statistik (2024), total populasi kendaraan bermotor di tanah air telah menembus angka 166 juta unit. Lonjakan jumlah kendaraan ini berdampak langsung pada meningkatnya kepadatan lalu lintas dan potensi insiden kecelakaan di berbagai ruas jalan. Oleh karena itu, pemerintah melalui Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan telah menegaskan urgensi penyelenggaraan sistem lalu lintas yang menjamin keselamatan, keamanan, serta ketertiban bagi seluruh pengguna jalan.

Permasalahan lalu lintas menjadi semakin kompleks seiring dengan meningkatnya volume kendaraan tersebut, di mana kecelakaan jalan raya menjadi salah satu ancaman keselamatan yang paling serius. Kecelakaan lalu lintas tidak hanya menyebabkan kerugian materi, tetapi juga berdampak pada keselamatan jiwa, kemacetan berkepanjangan, serta terganggunya produktivitas ekonomi. Tantangan utama dalam penanganan insiden ini terletak pada keterlambatan informasi kejadian di lapangan. Dalam

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

praktiknya, sering kali kecelakaan baru teridentifikasi setelah adanya laporan dari masyarakat atau petugas di tempat kejadian, yang menyebabkan proses evakuasi dan respons darurat menjadi kurang optimal (Pei et al., 2023). Hal ini menunjukkan urgensi diperlukannya sistem pemantauan yang mampu melakukan deteksi kejadian secara otomatis dan *real-time* (Ghahremannezhad, 2023).

Saat ini, infrastruktur kamera *Closed-Circuit Television* (CCTV) telah terpasang secara masif di berbagai ruas jalan utama, baik di tingkat perkotaan maupun jalan tol. Namun, pemanfaatan kamera tersebut masih didominasi oleh metode pengawasan konvensional yang sangat bergantung pada observasi visual manusia. Pendekatan ini memiliki keterbatasan dalam hal efisiensi dan reliabilitas karena petugas harus memantau banyak layar secara bersamaan sehingga rentan terhadap kelelahan visual dan *human error*. Kondisi tersebut menyebabkan potensi CCTV sebagai sumber data analitik yang kaya informasi belum dapat dimanfaatkan secara optimal untuk mendukung manajemen lalu lintas yang responsif (Husein et al., 2024).

Perkembangan pesat teknologi *Computer Vision* dan *Deep Learning* membuka peluang untuk mengatasi keterbatasan tersebut. *Computer Vision* memungkinkan sistem mengekstraksi informasi bermakna dari data video secara kontinu, seperti mendeteksi objek, melacak pergerakan kendaraan, hingga mengidentifikasi pola perilaku yang tidak wajar (Ikhsal et al., 2023). Algoritma deteksi objek berbasis *Convolutional Neural Network* (CNN), seperti YOLO, terus mengalami perkembangan dan telah banyak digunakan dalam berbagai kebutuhan *object detection* (Wang & Liang, 2024). Penerapan YOLOv8 juga telah dikembangkan untuk kebutuhan *real-time video surveillance*, salah satunya melalui pendekatan *attention based transformer-YOLOv8* yang menunjukkan pemanfaatan YOLOv8 pada lingkungan video surveilans (Nimma et al., 2025). Dalam konteks pemantauan lalu lintas, teknologi tersebut dapat digunakan untuk melakukan *vehicle counting*, estimasi kecepatan, serta deteksi insiden tabrakan (Nurdin et al., 2023; Perkasa et al., 2024). Namun, penggunaan YOLO, *tracking*, *vehicle counting*, dan estimasi kecepatan secara individual telah banyak diterapkan pada penelitian terdahulu sehingga penggunaan metode tersebut secara terpisah tidak menjadi aspek kebaruan utama penelitian ini.

Pengembangan sistem berbasis *web* menjadi pilihan arsitektur yang relevan karena memungkinkan proses penarikan *stream* video, ekstraksi data analitik, deteksi anomali, dan visualisasi informasi diintegrasikan dalam satu dasbor terpusat. Dalam penelitian ini, sistem memanfaatkan layanan akses data *live streaming* CCTV dari instansi terkait di bidang pengawasan lalu lintas sebagai salah satu sumber *input* visual, selain mendukung input berupa video untuk kebutuhan pengujian. Model YOLOv8n dioptimasi melalui 36 kombinasi *hyperparameter* pada *platform* Google Colab yang terdiri atas variasi *epoch*, *batch size*, optimizer SGD dan Adam, serta *learning rate* 0,001 dan 0,01. Model hasil optimasi kemudian diintegrasikan dengan DeepSORT untuk mempertahankan identitas kendaraan melalui *Tracking ID*, yang selanjutnya digunakan dalam *vehicle counting* dan estimasi kecepatan. Sistem juga menyediakan pengaturan *confidence threshold* melalui antarmuka web, sehingga pengguna dapat menyesuaikan tingkat sensitivitas deteksi sesuai dengan kondisi visual sumber video, seperti kondisi normal, dan kepadatan kendaraan Ketika kondisi visual tertentu menyebabkan munculnya lebih banyak deteksi dengan tingkat keyakinan rendah, pengguna dapat meningkatkan *confidence threshold* untuk menyaring sebagian deteksi tersebut. Sebaliknya, pada kondisi ketika objek sulit dikenali dan menghasilkan nilai *confidence* rendah, threshold dapat diturunkan agar sistem lebih sensitif terhadap keberadaan objek. Pengaturan ini digunakan sebagai mekanisme penyesuaian hasil inferensi dan bukan sebagai pengganti proses pelatihan model. Sistem yang dikembangkan mampu melakukan inferensi *real-time* dengan kecepatan rata-rata sekitar 25 *Frame per Second*.

Berdasarkan kondisi tersebut, penelitian ini tidak mengusulkan algoritma baru pada masing-masing komponen, tetapi menitikberatkan kebaruan pada integrasi fungsi analisis lalu lintas dalam satu *data pipeline* berbasis web yang dapat digunakan secara operasional dan disesuaikan oleh pengguna. YOLOv8n digunakan untuk mendeteksi kendaraan dan kecelakaan, DeepSORT mempertahankan identitas kendaraan, *Tracking ID* menjadi dasar untuk menghubungkan proses *vehicle counting* dan estimasi kecepatan, sedangkan hasil deteksi kecelakaan diintegrasikan dengan mekanisme notifikasi dan riwayat kejadian. Selain itu, penggunaan sumber *live streaming* CCTV dari layanan instansi terkait dan pengaturan *confidence threshold* secara langsung melalui antarmuka web memberikan fleksibilitas dalam menyesuaikan sensitivitas deteksi terhadap karakteristik sumber video. Dengan demikian, kontribusi penelitian ini terletak pada pengembangan sistem pemantauan lalu lintas terpadu yang menggabungkan deteksi kecelakaan, *tracking*, penghitungan kendaraan, estimasi kecepatan,

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

penyesuaian sensitivitas deteksi, dan visualisasi secara *real-time* dalam satu platform. Sistem tersebut diharapkan dapat mengurangi ketergantungan pada pengamatan manual serta mendukung proses pemantauan dan respons terhadap kondisi lalu lintas secara lebih efektif.

## KAJIAN LITERATUR

Untuk memahami kedudukan serta kontribusi orisinalitas penelitian ini, penulis melakukan peninjauan sistematis terhadap sejumlah riset terdahulu yang relevan. Kajian ini difokuskan pada pemetaan evolusi teknologi *Computer Vision* dalam ranah pemantauan lalu lintas, evaluasi terhadap kelebihan dan limitasi studi-studi sebelumnya, serta identifikasi celah penelitian (*research gap*) yang menjadi fondasi pengembangan sistem ini.

Ikhsal et al. (2023) melakukan riset terkait deteksi kecelakaan lalu lintas melalui rekaman CCTV dengan mengombinasikan metode peningkatan resolusi citra *Real-ESRGAN* serta model klasifikasi CNN. Hasil eksperimen menunjukkan bahwa teknik restorasi citra memberikan dampak signifikan dalam meningkatkan akurasi deteksi, khususnya pada rekaman CCTV dengan kualitas visual yang rendah. Namun, cakupan penelitian tersebut terbatas pada aspek identifikasi kecelakaan saja dan belum mengintegrasikan parameter analitik lalu lintas lainnya, seperti estimasi kecepatan maupun perhitungan volume kendaraan.

Husein et al. (2024) mengusulkan sistem deteksi insiden lalu lintas berbasis *deep learning* yang bekerja secara *real-time* pada video surveilans. Penelitian ini menggarisbawahi pentingnya pendeteksian dini untuk meminimalisasi risiko kemacetan susulan dan mempercepat respons pertolongan darurat. Meski demikian, arsitektur sistem yang dibangun hanya berfokus pada pengenalan kejadian kecelakaan tanpa menyajikan data kuantitatif lalu lintas, seperti jumlah kendaraan atau kecepatan arus lalu lintas yang krusial untuk evaluasi manajemen jalan.

Nurdin et al. (2023) menerapkan fusi antara algoritma deteksi objek YOLOv4 dan algoritma pelacakan *DeepSORT* untuk menganalisis pergerakan kendaraan dalam video lalu lintas. Pendekatan ini terbukti efektif dalam memetakan interaksi antar kendaraan serta mampu mengidentifikasi potensi tabrakan dengan tingkat akurasi yang memadai. Walaupun metodologi tersebut unggul dalam aspek pelacakan, riset ini belum mengeksplorasi estimasi kecepatan kendaraan secara mendalam sebagai komponen dari analisis karakteristik lalu lintas yang komprehensif.

Cai et al. (2024) mengeksplorasi penggunaan model *deep learning* untuk mengekstraksi data karakteristik lalu lintas, termasuk estimasi kecepatan dan volume kendaraan, dari data video CCTV. Penelitian tersebut menunjukkan tingkat akurasi yang lebih unggul dibandingkan dengan metode konvensional yang masih menggunakan sensor fisik di permukaan jalan. Sayangnya, model yang dikembangkan belum mampu menghubungkan hasil perhitungan kecepatan tersebut dengan deteksi kejadian kecelakaan yang terjadi di ruang lingkup pemantauan.

Perkasa et al. (2024) serta Adhisatria (2025) memberikan kontribusi penting dalam pembuktian efektivitas algoritma YOLOv8 untuk melakukan pendeteksian dan penghitungan jumlah kendaraan (*vehicle counting*) secara *real-time* dengan performa yang stabil. Lebih jauh, Husein et al. (2024) juga menegaskan bahwa integrasi fitur *multi-vehicle tracking* menjadi faktor determinan dalam menjaga tingkat akurasi penghitungan, terutama saat menghadapi kepadatan arus lalu lintas yang tinggi. Kendati demikian, seluruh studi tersebut masih memperlakukan fungsi penghitungan volume dan deteksi insiden sebagai entitas sistem yang terpisah, tanpa integrasi sistemik dalam satu alur pemrosesan.

Berdasarkan hasil analisis terhadap penelitian-penelitian terdahulu, dapat diketahui bahwa pemanfaatan Computer Vision dalam pemantauan lalu lintas telah berkembang pada berbagai aspek, mulai dari deteksi kecelakaan, estimasi kecepatan, hingga penghitungan jumlah kendaraan. Namun, sebagian besar penelitian masih mengembangkan ketiga fungsi tersebut secara terpisah sehingga belum mampu menghasilkan sistem pemantauan lalu lintas yang terintegrasi. Selain itu, masih terdapat perbedaan pendekatan dan metode yang digunakan dalam setiap penelitian, sehingga diperlukan pemahaman terhadap konsep-konsep dasar yang menjadi landasan dalam pengembangan sistem terpadu. Oleh karena itu, penelitian ini mengacu pada beberapa teori utama yang meliputi Computer Vision, algoritma YOLOv8, object tracking, serta estimasi kecepatan dan vehicle counting sebagai dasar dalam perancangan sistem yang diusulkan.

### Computer Vision

Computer Vision merupakan cabang dari Artificial Intelligence yang memungkinkan komputer memperoleh, memahami, dan menganalisis informasi dari citra maupun video secara otomatis. Dalam bidang transportasi cerdas (*Intelligent Transportation System*), Computer Vision banyak dimanfaatkan untuk

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

mendeteksi objek kendaraan, melacak pergerakan kendaraan, menghitung volume lalu lintas, mengestimasi kecepatan, hingga mendeteksi kejadian abnormal seperti kecelakaan. Pemanfaatan teknologi ini memungkinkan proses pemantauan lalu lintas dilakukan secara otomatis menggunakan kamera CCTV tanpa bergantung pada pengamatan manusia.

**YOLOv8**

YOLOv8 merupakan algoritma *object detection* berbasis *deep learning* yang mampu mendeteksi objek secara *real-time* dengan tingkat akurasi dan efisiensi komputasi yang tinggi. Perkembangan keluarga YOLO menunjukkan bahwa pendekatan *single-stage object detection* terus dikembangkan untuk memperoleh keseimbangan antara akurasi dan kecepatan inferensi pada berbagai kebutuhan deteksi objek (Wang & Liang, 2024). Dibandingkan generasi sebelumnya, YOLOv8 memiliki arsitektur yang lebih ringan dan mampu menghasilkan proses inferensi yang lebih cepat sehingga sesuai digunakan pada aplikasi *Computer Vision* yang memerlukan respons waktu nyata. Dalam penelitian ini, YOLOv8n digunakan sebagai model utama untuk mendeteksi kendaraan maupun kejadian kecelakaan pada video CCTV.Object tracking merupakan proses pelacakan objek yang telah terdeteksi agar setiap kendaraan memiliki identitas unik selama berada pada video. Melalui proses pelacakan tersebut, sistem dapat menghitung jumlah kendaraan menggunakan virtual line sekaligus mengestimasi kecepatan berdasarkan perpindahan posisi kendaraan pada setiap frame video. Integrasi antara object detection dan object tracking memungkinkan sistem melakukan analisis lalu lintas secara lebih akurat karena setiap kendaraan hanya dihitung satu kali serta dapat dipantau pergerakannya secara berkelanjutan.

Estimasi kecepatan kendaraan merupakan proses perhitungan kecepatan berdasarkan perubahan posisi kendaraan terhadap waktu selama proses pelacakan. Secara umum, kecepatan kendaraan dihitung menggunakan persamaan:

$$v = \frac{s}{t}$$

dengan:

- **v** = kecepatan kendaraan,
- **s** = jarak perpindahan kendaraan,
- **t** = waktu tempuh kendaraan.

Dalam sistem berbasis video, nilai perpindahan kendaraan diperoleh dari perubahan posisi objek pada setiap *frame* hasil pelacakan DeepSORT, sedangkan waktu diperoleh berdasarkan selisih jumlah *frame* terhadap nilai *Frame Per Second* (FPS) video. Melalui pendekatan tersebut, sistem dapat mengestimasi kecepatan kendaraan secara kontinu selama objek masih berada pada area pengamatan. Informasi kecepatan ini dimanfaatkan sebagai salah satu parameter untuk menganalisis karakteristik arus lalu lintas sekaligus mendukung proses identifikasi indikasi kecelakaan berdasarkan perubahan kecepatan kendaraan yang tidak normal.

Penghitungan jumlah kendaraan (*vehicle counting*) dilakukan menggunakan metode virtual line counting. Setiap kendaraan yang memiliki Unique ID akan dihitung ketika melintasi garis virtual yang telah ditentukan, sehingga kendaraan yang sama tidak akan dihitung lebih dari satu kali. Pendekatan ini mampu mengurangi terjadinya *double counting* serta menghasilkan informasi jumlah kendaraan yang lebih akurat. Integrasi antara deteksi objek, pelacakan kendaraan, estimasi kecepatan, dan *vehicle counting* memungkinkan sistem menyajikan informasi kondisi lalu lintas secara komprehensif dalam satu proses analisis *real-time*.

Oleh karena itu, kebaruan (*novelty*) dalam penelitian ini terletak pada pengembangan arsitektur cerdas berbasis YOLOv8n yang tidak hanya mengotomatisasi pendeteksian kecelakaan, estimasi kecepatan, dan penghitungan volume kendaraan secara bersamaan, tetapi juga mengoptimalkan tingkat presisi model melalui 36 kombinasi *hyperparameter tuning* di *platform* Google Colab. Dengan mengintegrasikan sistem ke dalam antarmuka *web* yang mampu memproses *input* visual sumber *live streaming* CCTV melalui RSTP secara *real-time* dengan kecepatan 25 FPS, penelitian ini menjadi solusi sistematis yang efektif bagi instansi terkait dalam mengoptimalkan pemantauan lalu lintas yang lebih akurat, responsif, dan terpadu.

**METODE PENELITIAN**

Penelitian ini merupakan riset terapan yang mengadopsi metodologi pengembangan sistem *Waterfall* untuk membangun platform pemantauan lalu lintas cerdas berbasis *Computer Vision*. Pemilihan model *Waterfall* didasarkan pada kebutuhan alur kerja yang sistematis dan terstruktur, mulai dari identifikasi kebutuhan fungsional, perancangan arsitektur sistem, implementasi model *deep learning*, pengujian performa, hingga pemeliharaan sistem. Permasalahan yang melatarbelakangi penelitian ini adalah tingginya beban kerja visual operator dalam memantau rekaman Closed-Circuit Television (CCTV) secara manual yang berpotensi menyebabkan *human error* dan keterlambatan respons saat terjadi kecelakaan. Sebagai

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

solusi, penelitian ini mengintegrasikan model deteksi objek YOLOv8n yang dioptimasi melalui 36 kombinasi *hyperparameter* pada *platform* Google Colab, dengan variasi *epoch* 50, 100, dan 150; *batch size* 16, 32, dan 64; optimizer Stochastic Gradient Descent (SGD) dan Adaptive Moment Estimation (Adam); serta *learning rate* 0,001 dan 0,01.

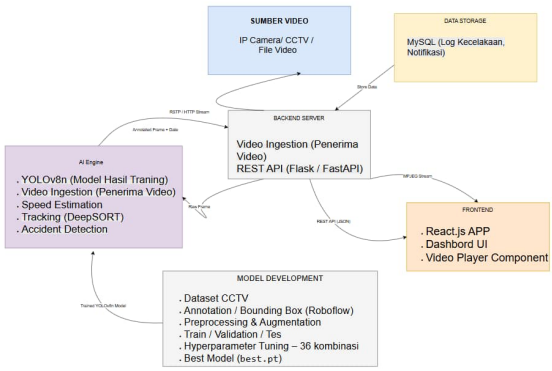
Dataset penelitian terdiri atas 1.071 citra hasil ekstraksi *frame* rekaman CCTV jalan raya dengan lima kelas objek, yaitu kecelakaan (*accident*), mobil (*car*), sepeda motor (*motorcycle*), truk (*truck*), dan bus (*bus*). Dataset dibagi menjadi 80% data *training*, 10% *validation*, dan 10% *testing*. Proses anotasi dilakukan menggunakan Roboflow dengan memberikan *bounding box* pada setiap objek sesuai kelasnya. Jumlah objek teranotasi terdiri atas 244 bus, 559 kecelakaan, 1.616 mobil, 1.975 sepeda motor, dan 718 truk, dengan total 5.112 objek. Data diperoleh dari rekaman CCTV dengan variasi kondisi pencahayaan, kepadatan kendaraan, ukuran objek, posisi kendaraan terhadap kamera, dan kondisi *occlusion*. Pada data *training* diterapkan *data augmentation* sebanyak tiga keluaran untuk meningkatkan variasi data, dengan perubahan *brightness* −15% hingga +15%, *blur* hingga 2,5 piksel, dan *noise* hingga 0,1% piksel.

Proses ekstraksi data analitik dari rekaman video diimplementasikan melalui pipa pemrosesan (*data pipeline*) yang terstruktur. Tahap awal meliputi *preprocessing* video dan ekstraksi *frame*, kemudian *frame* diproses oleh YOLOv8n untuk mendeteksi kendaraan dan kecelakaan. Hasil deteksi kendaraan dipadukan dengan algoritma DeepSORT untuk memberikan ID unik pada setiap objek sehingga dapat mengurangi penghitungan ganda ketika kendaraan bergerak pada area pantau. Estimasi kecepatan kendaraan dilakukan berdasarkan perubahan posisi *centroid* terhadap waktu tempuh antar-*frame*, sedangkan penghitungan volume kendaraan menerapkan metode *virtual line counting* berdasarkan ID unik yang melintasi garis batas yang telah ditentukan.

Komponen deteksi kecelakaan dikembangkan melalui analisis kondisi objek dan pergerakannya, dengan memonitor anomali seperti deselerasi secara mendadak, kendaraan yang berhenti tidak wajar, serta tumpang tindih (*overlap*) antar-*bounding box* yang dapat mengindikasikan kontak antar-kendaraan. Seluruh hasil analisis berupa statistik kendaraan dan notifikasi kecelakaan diintegrasikan ke dalam dashboard berbasis *web* yang menyajikan informasi secara *real-time* dengan kecepatan pemrosesan sekitar 25 Frame Per Second. Sistem juga mengintegrasikan sumber *live streaming* CCTV melalui RSTP sehingga hasil deteksi, pelacakan, estimasi kecepatan, penghitungan kendaraan, dan informasi kecelakaan dapat dipantau secara terpadu.

**Arsitektur Sistem**

Arsitektur sistem dikembangkan menggunakan model *client–server* yang mengintegrasikan Sumber Video, Backend Server, Model Development, AI Engine, Frontend, dan Data Storage. Sumber video berupa IP Camera/CCTV maupun file video mengirimkan *stream* ke Backend Server yang dibangun menggunakan Flask/FastAPI sebagai pusat penerimaan dan pengelolaan video serta penyedia REST API. Sebelum digunakan pada proses inferensi, model YOLOv8n dikembangkan melalui *Model Development* yang meliputi penyusunan dataset CCTV, anotasi *bounding box* menggunakan Roboflow, *preprocessing* dan *augmentation*, pembagian data *train/validation/test*, pelatihan menggunakan 36 kombinasi *hyperparameter*, serta pemilihan *best model* (best.pt). Model terlatih tersebut kemudian digunakan pada AI Engine untuk melakukan *video ingestion*, deteksi objek menggunakan YOLOv8n, *tracking* menggunakan DeepSORT, estimasi kecepatan, *vehicle counting*, dan deteksi kecelakaan. Hasil analisis dari AI Engine dikirimkan kembali melalui Backend Server ke Frontend dalam bentuk *MJPEG stream* dan REST API sehingga dapat ditampilkan secara *real-time* pada aplikasi React.js melalui *browser*. Informasi kejadian kecelakaan dan notifikasi selanjutnya disimpan pada Data Storage berbasis MySQL untuk kebutuhan pencatatan dan riwayat kejadian.



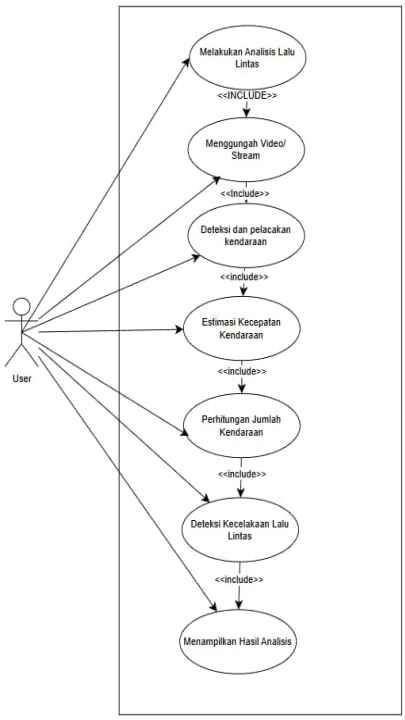

**Gambar 1.** Diagram Arsitektur Sistem

**Perancangan Sistem**

Perancangan sistem digambarkan menggunakan Unified Modeling Language (UML) berupa Use Case Diagram untuk menggambarkan interaksi antara pengguna dan sistem.

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

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




**Gambar 2.** Use Case Diagram

Penjelasan gambar:

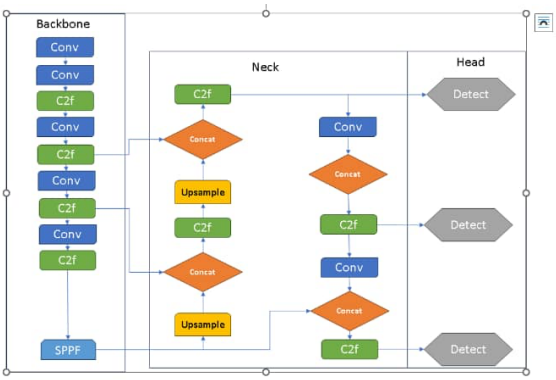
1. User merupakan aktor utama yang berinteraksi dengan seluruh fungsi sistem. User bertugas mengoperasikan sistem mulai dari memasukkan sumber video CCTV hingga melihat hasil analisis yang dihasilkan oleh sistem secara real-time.
2. Melakukan Analisis Lalu Lintas merupakan proses utama yang dijalankan oleh pengguna untuk memulai proses analisis video CCTV. Use case ini menginisialisasi seluruh tahapan analisis sehingga sistem dapat memproses data masukan secara otomatis.
3. Mengunggah Video/Stream merupakan use case yang digunakan untuk memasukkan sumber data berupa file video CCTV maupun *live stream* dari kamera. Setelah video berhasil dimuat dan divalidasi, sistem akan memulai proses ekstraksi frame sebagai masukan bagi modul Computer Vision.
4. Deteksi dan Pelacakan Kendaraan merupakan proses inti yang menggunakan model YOLOv8 untuk mendeteksi kendaraan pada setiap frame video. Selanjutnya sistem melakukan object tracking dengan memberikan identitas (*ID*) unik pada setiap kendaraan sehingga posisi dan lintasan kendaraan dapat dipantau secara berkelanjutan selama proses analisis berlangsung.
5. Estimasi Kecepatan Kendaraan merupakan proses lanjutan yang memanfaatkan hasil pelacakan kendaraan untuk menghitung kecepatan berdasarkan perpindahan posisi objek pada setiap frame video. Informasi kecepatan digunakan sebagai salah satu parameter dalam menganalisis kondisi lalu lintas dan mendukung proses deteksi kecelakaan.
6. Perhitungan Jumlah Kendaraan dilakukan dengan memanfaatkan *ID* unik hasil pelacakan kendaraan. Sistem akan menghitung kendaraan ketika melewati garis virtual (virtual line) sehingga setiap kendaraan hanya dihitung satu kali. Hasil penghitungan ini digunakan untuk memperoleh informasi volume lalu lintas secara otomatis.
7. Deteksi Kecelakaan Lalu Lintas merupakan proses analisis yang menggabungkan informasi hasil deteksi kendaraan, pelacakan, dan estimasi kecepatan untuk mengidentifikasi indikasi kecelakaan. Sistem menganalisis pola pergerakan kendaraan, seperti perubahan kecepatan secara drastis, berhenti mendadak, maupun terjadinya tabrakan atau tumpang tindih (*overlap*) antar kendaraan, kemudian memberikan notifikasi apabila terindikasi terjadi kecelakaan.
8. Menampilkan Hasil Analisis merupakan tahapan akhir dari proses sistem. Seluruh hasil analisis berupa video yang telah diberi anotasi (*bounding box*), estimasi kecepatan kendaraan, jumlah kendaraan, serta informasi indikasi kecelakaan ditampilkan pada antarmuka web sehingga pengguna dapat melakukan pemantauan lalu lintas secara real-time.

## HASIL DAN PEMBAHASAN

### Pembuatan Model YOLOv8n

Pembuatan model YOLOv8n diawali dengan proses penyusunan dataset menggunakan platform Roboflow. Dataset yang digunakan terdiri atas 1.071 citra yang diperoleh dari hasil ekstraksi rekaman CCTV jalan raya sehingga karakteristik objek yang digunakan sesuai dengan kondisi implementasi sistem. Selanjutnya dilakukan proses *annotation* menggunakan Roboflow dengan memberikan label pada setiap objek sesuai kelasnya, yaitu kecelakaan (accident), mobil (car), sepeda motor (motorcycle), truk (truck), dan bus (bus). Setelah seluruh objek selesai diberi anotasi, dilakukan tahap *preprocessing* dan *augmentation* untuk meningkatkan variasi data pelatihan sebelum dataset diekspor ke dalam format YOLOv8.

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




**Gambar 3.** Diagram Arsitektur YOLOv8

Proses *augmentation* dilakukan menggunakan fitur yang tersedia pada Roboflow dengan menerapkan perubahan brightness pada rentang −15% hingga +15%, blur hingga 2,5 piksel, serta penambahan noise hingga 0,1% piksel. Penerapan *augmentation* bertujuan untuk meningkatkan keragaman data sehingga model lebih mampu mengenali objek pada berbagai kondisi pencahayaan dan kualitas citra CCTV.

Pelatihan model dilakukan pada platform Google Colab menggunakan framework Ultralytics YOLOv8. Untuk memperoleh model dengan performa terbaik dilakukan pengujian terhadap 36 kombinasi *hyperparameter* yang diperoleh dari variasi jumlah *epoch*, *batch size*, *optimizer*, dan *learning rate*. Variasi parameter yang digunakan pada penelitian ini ditunjukkan pada Tabel 1.

**Tabel 1.** Konfigurasi Evaluasi Model YOLOv8n

| **Parameter** | **Nilai** |
|---|---|
| Dataset | 1.071 Citra |
| Kelas | Kecelakaan, Mobil, Motor, Truk , Bus |
| Model | YOLOv8n |
| Image Size | 640 × 640 |
| Epoch | 50, 100, 150 |
| Batch Size | 16, 32, 64 |
| Optimizer | SGD, Adam |
| Learning Rate | 0.01, 0.001 |
| Framework | Ultralytics YOLOv8 |
| Jumlah Kombinasi | 36 |

Setiap kombinasi *hyperparameter* menghasilkan performa model yang berbeda. Oleh karena itu, seluruh 36 kombinasi dievaluasi menggunakan Precision, Recall, F1-Score, mAP@50, dan mAP@50–95. Berdasarkan hasil pengujian, dilakukan perbandingan antara konfigurasi dengan performa mAP@50 tertinggi menggunakan optimizer SGD dan konfigurasi terbaik menggunakan optimizer Adam. Perbandingan tersebut dilakukan untuk melihat tidak hanya tingkat performa deteksi, tetapi juga kestabilan proses pelatihan dan indikasi *overfitting* pada masing-masing optimizer.

**Tabel 2.** Perbandingan Model terbaik SGD dan Adam

| **Kombinasi** | **Precision** | **Recall** | **F1-Score** | **mAP50** |
|---|---|---|---|---|
| epoch50_bs32_ SGD_lr0.001 | 0.8306 | 0.7633 | 0.7955 | 0.8623 |
| epoch50_bs16_ Adam_lr0.01 | 0.7736 | 0.7942 | 0.7838 | 0.8409 |

Berdasarkan Tabel 2, konfigurasi menggunakan optimizer SGD menghasilkan nilai mAP@50 sebesar 0,8623, lebih tinggi dibandingkan konfigurasi Adam yang memperoleh nilai mAP@50 sebesar 0,8409. Model SGD juga memperoleh Precision sebesar 0,8306, sedangkan model Adam memperoleh Precision sebesar 0,7736. Sebaliknya, nilai Recall model Adam sebesar 0,7942 lebih tinggi dibandingkan model SGD sebesar 0,7633. Perbedaan tersebut menunjukkan bahwa kedua optimizer memiliki karakteristik performa yang berbeda dalam menghasilkan prediksi objek.

Hasil evaluasi tersebut tidak digunakan dengan hanya mempertimbangkan nilai mAP@50. Berdasarkan kurva pelatihan dan validasi, model SGD menunjukkan indikasi *overfitting* pada tahap akhir pelatihan, sedangkan model Adam menunjukkan proses pelatihan yang lebih stabil dan tidak menunjukkan indikasi *overfitting* yang sama. Dengan demikian, meskipun model SGD memperoleh mAP@50 lebih tinggi, nilai tersebut perlu dianalisis bersama dengan Precision, Recall, F1-Score, serta karakteristik proses pelatihan dan validasi.

**Tabel 3.** Hasil Performa Model YOLOv8n

| **Metrik** | **Nilai** |
|---|---|
| Precision | 0.8306 |
| Recall | 0.7633 |
| mAP@50 | 0.8623 |
| mAP@50-95 | 0.517 |
| mAP@50 Kelas Kecelakaan | 0.881 |

Hasil evaluasi menunjukkan bahwa model YOLOv8n memiliki kemampuan deteksi yang baik untuk mendukung implementasi sistem analisis lalu lintas. Nilai Precision sebesar 0,8306 menunjukkan bahwa sebagian besar objek yang dideteksi merupakan objek yang benar sehingga jumlah *false positive* dapat diminimalkan. Sementara itu, nilai Recall sebesar

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

0,7633 menunjukkan bahwa model mampu mendeteksi sebagian besar objek pada data pengujian meskipun masih terdapat beberapa objek yang belum berhasil dikenali. Nilai F1-Score sebesar 0,7955 menunjukkan keseimbangan antara Precision dan Recall pada model yang digunakan.

Nilai mAP@50 sebesar 0,8623 menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi lokasi objek pada ambang *Intersection over Union* (IoU) sebesar 0,5. Selain itu, nilai mAP@50 sebesar 0,881 pada kelas kecelakaan menunjukkan bahwa model memiliki performa yang baik dalam mengenali kejadian kecelakaan pada metrik tersebut. Nilai mAP@50–95 sebesar 0,517 juga menunjukkan bahwa performa deteksi menjadi lebih rendah ketika kriteria IoU dibuat lebih ketat. Hal ini menunjukkan bahwa model memiliki kemampuan deteksi objek yang baik, tetapi ketepatan lokalisasi *bounding box* masih dapat mengalami penurunan pada ambang IoU yang lebih tinggi.

Berdasarkan hasil perbandingan, model SGD tetap dipilih dan digunakan pada tahap implementasi sistem karena menghasilkan nilai mAP@50 tertinggi dari konfigurasi yang diuji. Pemilihan tersebut tidak hanya mempertimbangkan mAP, tetapi juga Precision, Recall, F1-Score, serta hasil analisis proses pelatihan dan validasi. Meskipun terdapat indikasi *overfitting* pada model SGD, performanya pada data pengujian tetap menunjukkan kemampuan deteksi yang sesuai dengan kebutuhan sistem. Sementara itu, model Adam yang memiliki proses pelatihan lebih stabil digunakan sebagai pembanding untuk menunjukkan bahwa nilai mAP yang lebih tinggi perlu dianalisis bersama aspek kestabilan dan generalisasi model.

Dengan demikian, proses optimasi melalui 36 kombinasi *hyperparameter* memberikan dasar untuk memilih konfigurasi yang sesuai dengan kebutuhan implementasi. Model SGD yang dipilih selanjutnya diintegrasikan dengan algoritma DeepSORT untuk melakukan pelacakan kendaraan, estimasi kecepatan, penghitungan jumlah kendaraan, dan deteksi kecelakaan pada aplikasi berbasis web. Integrasi tersebut menjadi dasar dalam implementasi sistem analisis lalu lintas secara *real-time*.

**Implementasi Sistem**

Sistem yang dikembangkan merupakan aplikasi berbasis web yang mengintegrasikan model YOLOv8n untuk melakukan deteksi kecelakaan, estimasi kecepatan kendaraan, dan penghitungan jumlah kendaraan secara *real-time*. Sistem dibangun menggunakan arsitektur *client-server*, di mana seluruh proses inferensi model dilakukan pada sisi server, sedangkan pengguna mengakses hasil analisis melalui antarmuka web melalui *browser*. Pendekatan ini memungkinkan proses komputasi dilakukan secara terpusat sehingga perangkat pengguna hanya berfungsi sebagai media untuk mengelola sumber video dan menampilkan hasil analisis. Dengan demikian, kebutuhan sumber daya komputasi pada perangkat pengguna menjadi lebih ringan, sementara proses analisis tetap dapat berjalan secara optimal pada server.

Tahap awal penggunaan sistem dilakukan melalui halaman Input Media, yang berfungsi sebagai antarmuka untuk menentukan sumber data yang akan dianalisis. Sistem mendukung dua jenis masukan, yaitu video lokal dengan format MP4, AVI, dan MKV, serta *live streaming* CCTV menggunakan URL RTSP. Fleksibilitas sumber masukan ini memungkinkan sistem digunakan baik untuk pengujian menggunakan dataset video maupun implementasi langsung pada kamera pengawas di lapangan. Pengguna dapat memilih sumber video sesuai kebutuhan tanpa melakukan perubahan terhadap konfigurasi sistem sehingga proses analisis menjadi lebih praktis dan efisien.

Selain memilih sumber video, pengguna juga dapat menambahkan informasi kamera CCTV yang meliputi nama lokasi, alamat RTSP, koordinat *latitude*, dan *longitude*. Informasi tersebut disimpan ke dalam basis data sehingga konfigurasi kamera dapat digunakan kembali tanpa perlu dilakukan pengaturan ulang pada setiap proses analisis. Fitur ini mempermudah pengelolaan beberapa titik kamera sekaligus serta mendukung implementasi sistem pada jaringan CCTV yang terdiri dari banyak lokasi pemantauan. Penyimpanan informasi lokasi juga memberikan kemudahan dalam proses identifikasi sumber video ketika sistem mendeteksi kejadian kecelakaan maupun ketika operator melakukan peninjauan terhadap riwayat hasil deteksi.

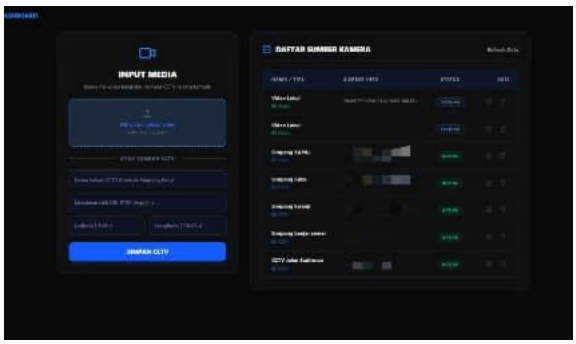
Setelah pengguna memilih video atau kamera CCTV, sistem akan mengirimkan sumber video ke server untuk diproses menggunakan model YOLOv8n. Pada tahap ini, setiap *frame* video dianalisis secara berurutan untuk mendeteksi objek kendaraan dan objek kecelakaan. Selanjutnya, hasil deteksi diteruskan ke algoritma DeepSORT untuk melakukan proses pelacakan objek sehingga setiap kendaraan memperoleh identitas unik (*Unique ID*) yang dipertahankan selama kendaraan masih berada pada area pengamatan.

Melalui integrasi antara YOLOv8n dan DeepSORT, sistem tidak hanya mampu mendeteksi keberadaan kendaraan, tetapi juga menghasilkan informasi tambahan berupa estimasi kecepatan

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

kendaraan, jumlah kendaraan yang sedang berada pada area pemantauan (*Current Vehicle*), jumlah total kendaraan yang telah melintas (*Total Vehicle*), serta rata-rata kecepatan kendaraan (*Average Speed*). Seluruh hasil analisis tersebut kemudian dikirimkan kembali ke antarmuka web dan ditampilkan secara *real-time* pada dashboard sehingga pengguna dapat memantau kondisi lalu lintas tanpa harus melakukan pengamatan secara manual terhadap rekaman CCTV.

Implementasi sistem ini menunjukkan bahwa integrasi model deteksi objek dengan algoritma pelacakan mampu menghasilkan proses analisis lalu lintas yang berjalan secara otomatis. Seluruh proses, mulai dari pengambilan sumber video, deteksi objek, pelacakan kendaraan, estimasi kecepatan, penghitungan jumlah kendaraan, hingga deteksi kecelakaan, dilakukan dalam satu alur pemrosesan yang terintegrasi sehingga mempermudah proses pemantauan lalu lintas secara berkelanjutan.




**Gambar 4.** Halaman Inputan

**Implementasi Dashboard Analisis Lalu Lintas**

Dashboard merupakan halaman utama yang menampilkan hasil analisis video secara *real-time*. Seluruh informasi yang dihasilkan dari proses deteksi dan pelacakan objek ditampilkan secara langsung sehingga pengguna dapat memantau kondisi lalu lintas melalui satu antarmuka. Dashboard dirancang untuk menyajikan informasi utama yang dibutuhkan operator, mulai dari hasil deteksi kendaraan, estimasi kecepatan, jumlah kendaraan, hingga notifikasi kecelakaan.

Sistem menggunakan YOLOv8n untuk mendeteksi objek kendaraan pada setiap *frame* video, kemudian hasil deteksi diproses menggunakan algoritma DeepSORT untuk melakukan pelacakan objek (*multi-object tracking*). Melalui proses ini, setiap kendaraan diberikan Unique ID yang dipertahankan selama kendaraan berada pada area pengamatan. Pemberian identitas unik ini memungkinkan sistem membedakan setiap kendaraan meskipun terdapat banyak kendaraan yang bergerak secara bersamaan pada satu waktu.

Keberadaan Unique ID memiliki peran penting dalam menjaga konsistensi hasil analisis. Dengan mempertahankan identitas kendaraan pada setiap *frame*, sistem mampu menghindari terjadinya *double counting* ketika kendaraan masih berada pada area pemantauan. Selain itu, pelacakan yang stabil juga menjadi dasar dalam proses estimasi kecepatan kendaraan karena perubahan posisi kendaraan dapat diamati secara berkelanjutan selama proses analisis berlangsung.

Estimasi kecepatan kendaraan diperoleh berdasarkan perubahan posisi kendaraan pada setiap *frame* hasil pelacakan DeepSORT. Perubahan posisi tersebut digunakan untuk menghitung kecepatan kendaraan sehingga sistem dapat menampilkan nilai kecepatan masing-masing kendaraan dalam satuan km/jam. Informasi kecepatan ditampilkan secara langsung di atas *bounding box* kendaraan tanpa membedakan kelas kendaraan, sehingga pengguna dapat memantau kecepatan setiap kendaraan yang sedang melintas pada area pengamatan.

Selain menampilkan kecepatan setiap kendaraan, sistem juga menghitung Average Speed, yaitu rata-rata kecepatan seluruh kendaraan yang sedang terdeteksi pada area pengamatan. Informasi ini memberikan gambaran umum mengenai kondisi arus lalu lintas pada lokasi pemantauan. Nilai rata-rata kecepatan yang relatif tinggi menunjukkan bahwa kendaraan bergerak dengan lancar, sedangkan penurunan nilai rata-rata kecepatan dapat mengindikasikan adanya peningkatan kepadatan lalu lintas maupun gangguan pada ruas jalan. Dengan demikian, informasi estimasi kecepatan tidak hanya digunakan sebagai data individual kendaraan, tetapi juga berfungsi sebagai indikator kondisi lalu lintas secara keseluruhan.

Dashboard juga menampilkan informasi Current Vehicle, yaitu jumlah kendaraan yang sedang berada pada area pengamatan, serta Total Vehicle, yaitu jumlah akumulasi kendaraan yang telah melintas sejak proses analisis dimulai. Penghitungan dilakukan berdasarkan Unique ID yang dihasilkan oleh DeepSORT sehingga setiap kendaraan hanya dihitung satu kali meskipun terdeteksi pada banyak *frame*. Pendekatan ini meningkatkan akurasi proses penghitungan kendaraan dan mengurangi kemungkinan terjadinya kesalahan akibat kendaraan yang terdeteksi berulang.

Sebagai indikator performa sistem, dashboard turut menampilkan nilai Frame Per Second (FPS) selama proses inferensi berlangsung. Nilai FPS menunjukkan kemampuan sistem dalam memproses video secara *real-time*. Semakin tinggi nilai FPS yang

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

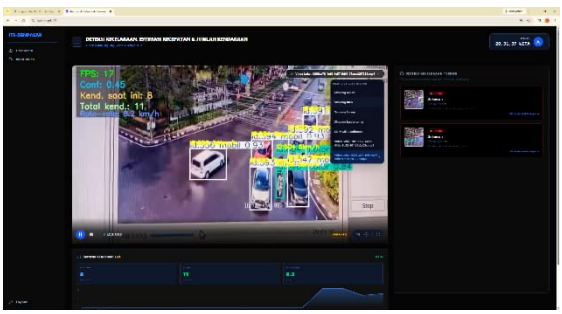
ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

dihasilkan, semakin cepat sistem melakukan analisis pada setiap *frame* video sehingga informasi yang ditampilkan kepada pengguna menjadi lebih responsif.

Selain menyajikan informasi analisis lalu lintas, dashboard juga mengintegrasikan fitur deteksi kecelakaan menggunakan model YOLOv8n. Sistem secara terus-menerus memantau hasil deteksi pada setiap *frame* video. Apabila objek kecelakaan terdeteksi secara konsisten selama minimal 2 detik, sistem akan menampilkan notifikasi berupa *pop-up* yang berisi tangkapan layar kejadian, waktu deteksi, lokasi CCTV, serta detail hasil deteksi. Mekanisme ini diterapkan untuk mengurangi kemungkinan munculnya notifikasi akibat deteksi sesaat (*false detection*) yang dapat terjadi pada beberapa *frame*.

Setelah notifikasi ditampilkan, pengguna diberikan pilihan untuk menyimpan hasil deteksi ke dalam basis data atau mengabaikannya apabila teridentifikasi sebagai false positive. Seluruh data yang disimpan akan masuk ke dalam menu Accident History sehingga riwayat kejadian kecelakaan dapat ditinjau kembali sewaktu-waktu. Fitur ini memudahkan operator dalam melakukan dokumentasi kejadian sekaligus mendukung proses evaluasi terhadap kondisi lalu lintas berdasarkan data historis yang telah tersimpan.

Secara keseluruhan, dashboard mampu mengintegrasikan hasil deteksi kendaraan, pelacakan objek, estimasi kecepatan, penghitungan jumlah kendaraan, dan deteksi kecelakaan ke dalam satu tampilan yang mudah dipahami. Integrasi berbagai informasi tersebut memberikan nilai tambah dibandingkan sistem yang hanya menampilkan hasil deteksi objek, karena pengguna dapat memperoleh informasi kondisi lalu lintas secara lebih komprehensif dalam satu proses analisis *real-time*.



**Gambar 5.** Halaman Dashboard

**Tabel 4.** Informasi Analisi Lalu Lintas Yang Ditampilkan Sistem

| **Komponen** | **Fungsi** |
|---|---|
| Bounding Box | Hasil deteksi kendaraan |
| Unique ID | Identitas kendaraan hasil DeepSORT |
| Speed | Estimasi kecepatan setiap kendaraan |
| Average Speed | Rata-rata kecepatan kendaraan |
| Current Vehicle | Jumlah kendaraan aktif |
| Total Vehicle | Total kendaraan yang telah melintas |
| FPS | Kecepatan pemrosesan sistem |
| Accident Notification | Pop-up deteksi kecelakaan |
| Accident History | Riwayat kejadian kecelakaan |

Berdasarkan implementasi dashboard yang telah dikembangkan, seluruh informasi analisis lalu lintas dapat ditampilkan secara *real-time* dalam satu antarmuka. Integrasi antara YOLOv8n sebagai model deteksi objek dan DeepSORT sebagai algoritma pelacakan memungkinkan sistem tidak hanya mendeteksi kendaraan, tetapi juga menghasilkan informasi estimasi kecepatan, jumlah kendaraan, serta notifikasi kecelakaan secara bersamaan. Hasil implementasi ini menunjukkan bahwa sistem mampu memberikan informasi kondisi lalu lintas yang lebih lengkap sehingga dapat membantu operator dalam melakukan pemantauan dan pengambilan keputusan secara lebih cepat

**Evaluasi Sistem**

Evaluasi dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai dengan rancangan. Hasil pengujian menunjukkan bahwa sistem mampu menerima masukan berupa video lokal maupun *live streaming* CCTV, melakukan deteksi kendaraan menggunakan YOLOv8n, pelacakan objek menggunakan DeepSORT, estimasi kecepatan, penghitungan jumlah kendaraan, serta deteksi kecelakaan secara *real-time*.

**Tabel 5.** Hasil Evaluasi Sistem

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil** |
|---|---|---|---|
| 1 | Input video lokal | Video berhasil diproses | Berhasil |
| 2 | Input RTSP CCTV | Streaming berhasil diproses | Berhasil |
| 3 | Deteksi kendaraan | Bounding box tampil | Berhasil |
| 4 | Pelacakan DeepSORT | Unique ID muncul | Berhasil |
| 5 | Estimasi kecepatan | Speed tampil | Berhasil |
| 6 | Current Vehicle | Jumlah kendaraan aktif tampil | Berhasil |

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

| | | | |
|---|---|---|---|
| 7 | Total Vehicle | Akumulasi kendaraan tampil | Berhasil |
| 8 | Average Speed | Nilai rata-rata tampil | Berhasil |
| 9 | Deteksi kecelakaan | Pop-up muncul saat kecelakaan | Berhasil |
| 10 | Riwayat kecelakaan | Data tersimpan ke database | Berhasil |

Berdasarkan hasil pengujian, integrasi YOLOv8n sebagai pendeteksi objek dan DeepSORT sebagai pelacak objek mampu menjalankan fungsi analisis lalu lintas secara *real-time*. Sistem dapat menampilkan informasi estimasi kecepatan, *Current Vehicle*, *Total Vehicle*, dan *Average Speed*, serta memberikan notifikasi ketika objek kecelakaan terdeteksi. Hasil pengujian ini menunjukkan bahwa fungsi-fungsi sistem telah berjalan sesuai rancangan, tetapi belum menunjukkan tingkat akurasi masing-masing fungsi. Oleh karena itu, evaluasi khusus diperlukan untuk mengukur performa deteksi kecelakaan secara kuantitatif.




**Gambar 6.** Tangkapan Layar Proses Pengujian

**Tabel 6.** Pengujian Aktual Sistem

| **Parameter** | **Pengujian 1** | **Pengujian 2** | **Pengujian 3** |
|---|---|---|---|
| Sumber Video | RTSP CCTV | RTSP CCTV | Video Lokal |
| Durasi Pengujian | 1 menit | 1 menit | 1 menit |
| Fokus Kamera | 1 jalur persimpangan | 1 jalur jalan | 1 jalur persimpangan |
| Kondisi Lalu Lintas | Sedang | Rendah | Tinggi |
| FPS Rata-rata | 26 FPS | 25 FPS | 30 FPS |
| Total Kendaraan Diamati | 34 | 29 | 53 |
| Kendaraan Berhasil Dideteksi | 29 | 24 | 45 |
| Rata-rata *Confidence* | 78,4% | 76,8% | 80,1% |
| Kesalahan Deteksi | 3 | 3 | 4 |
| Kendaraan Terlewat | 5 | 5 | 8 |
| Kecepatan Mnimun | 0 km/jam | 0 km/jam | 0 km/jam |
| Kecepatan Maksimum | 42 km/jam | 38 km/jam | 45 km/jam |
| Avarage Speed | 21,4 km/jam | 24,7 km/jam | 27,1 km/jam |
| Deteksi Kecelakaan | Tidak Ada | Tidak Ada | Ada |

Berdasarkan Tabel 6, sistem mampu menjalankan proses deteksi dan pelacakan kendaraan pada tiga sumber video dengan rentang 25–30 FPS. Pada pengujian, jumlah kendaraan yang diamati masing-masing sebesar 34, 29, dan 53 kendaraan, dengan 29, 24, dan 45 kendaraan berhasil dideteksi serta 5, 5, dan 8 kendaraan terlewat. Hasil tersebut menunjukkan bahwa DeepSORT dapat mempertahankan identitas kendaraan untuk mendukung proses *vehicle counting* dan analisis pergerakan kendaraan. Jumlah kendaraan yang terlewat dan kesalahan deteksi cenderung meningkat pada kondisi dengan jumlah objek yang lebih banyak, yang menunjukkan bahwa kepadatan kendaraan dapat memengaruhi proses deteksi dan pelacakan. Nilai *confidence* berada pada rentang 76,8%–80,1%. Sementara itu, estimasi kecepatan menghasilkan nilai minimum 0 km/jam karena kendaraan dapat berhenti pada persimpangan, nilai maksimum 38–45 km/jam, dan *Average Speed* 21,4–27,1 km/jam.

**Tabel 7.** Analisis Pengujian Estimasi Kecepatan Kendaraan

| **Parameter** | **Pengujian 1** | **Pengujian 2** | **Pengujian 3** |
|---|---|---|---|
| Posisi Kendaraan | Dekat kamera | Jarak menengah | Jauh dari kamera |
| Kecepatan *Speedometer* | 30 km/jam | 40 km/jam | 45 km/jam |
| Kecepatan Sistem | 29 km/jam | 35 km/jam | 32 km/jam |
| Selisih | 1 km/jam | 5 km/jam | 13 km/jam |
| Persentase Error | 3,33% | 12,50% | 28,89% |
| Hasil | Mendekati referensi | Sedikit menyimpang | Menyimpang |

Berdasarkan Tabel 7, hasil estimasi kecepatan menunjukkan bahwa ketepatan sistem dipengaruhi oleh jarak kendaraan terhadap kamera. Pada posisi dekat kamera, hasil estimasi sebesar 29 km/jam mendekati kecepatan referensi 30 km/jam dengan *error* 3,33%. Ketika kendaraan berada pada jarak menengah, *error* meningkat menjadi 12,50%, sedangkan pada posisi jauh dari kamera *error* meningkat menjadi 28,89%.

Peningkatan *error* tersebut disebabkan oleh semakin kecilnya perpindahan posisi kendaraan yang terlihat pada bidang citra ketika kendaraan menjauh dari kamera. Kondisi ini menyebabkan perubahan posisi *centroid* antar-*frame* menjadi lebih kecil sehingga perhitungan perpindahan dalam satuan piksel kurang merepresentasikan perpindahan sebenarnya di dunia nyata. Dengan demikian, sistem menghasilkan

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

estimasi kecepatan yang lebih mendekati nilai *speedometer* pada area pengamatan yang lebih dekat dengan kamera, sedangkan akurasinya menurun ketika objek berada semakin jauh.

Hasil tersebut juga menunjukkan bahwa estimasi kecepatan pada penelitian ini bersifat estimatif dan dipengaruhi oleh perspektif kamera serta posisi objek. Oleh karena itu, hasil kecepatan pada area yang jauh dari kamera perlu diinterpretasikan dengan mempertimbangkan keterbatasan perspektif dan resolusi objek

**Analisis Implementasi Sistem**

Implementasi sistem analisis lalu lintas berbasis YOLOv8n dan DeepSORT mampu mengintegrasikan deteksi kendaraan, estimasi kecepatan, penghitungan jumlah kendaraan, dan deteksi kecelakaan dalam satu aplikasi berbasis web. Seluruh proses analisis dilakukan secara otomatis dan hasilnya ditampilkan secara *real-time* pada dashboard.

Berdasarkan hasil pengujian 36 kombinasi *hyperparameter*, model yang dipilih menghasilkan mAP@50 sebesar 0,8623, mAP@50–95 sebesar 0,517, dan mAP@50 kelas kecelakaan sebesar 0,881. Nilai tersebut menunjukkan kemampuan deteksi yang baik pada dataset pengujian. Namun, mAP@50 kelas kecelakaan tidak dapat langsung dianggap sebagai akurasi deteksi kecelakaan pada video karena mAP mengukur performa deteksi objek pada dataset, sedangkan deteksi kecelakaan pada sistem merupakan proses berbasis video.

Integrasi DeepSORT memberikan *Unique ID* pada kendaraan sehingga identitas objek dapat dipertahankan antar-*frame*. Informasi tersebut digunakan sebagai dasar penghitungan kendaraan dan estimasi kecepatan serta membantu mengurangi *double counting*. Namun, kestabilan *tracking* dapat berkurang pada kondisi *occlusion*, kendaraan yang sangat berdekatan, atau objek berukuran kecil.

Pada implementasi CCTV, sistem menghasilkan FPS rata-rata sekitar 25 FPS, sedangkan pengujian video lokal mencapai 30 FPS. Perbedaan tersebut menunjukkan bahwa performa *real-time* dipengaruhi oleh sumber video dan beban pemrosesan. Oleh karena itu, nilai FPS merupakan hasil pada skenario pengujian tertentu dan tidak dapat digeneralisasikan untuk semua kondisi CCTV.

**Evaluasi Khusus Deteksi Kecelakaan**

Untuk melengkapi evaluasi mAP dan *black-box testing*, dilakukan evaluasi khusus terhadap mekanisme deteksi kecelakaan menggunakan 5 video kecelakaan dan 5 *live streaming* CCTV non-kecelakaan. Kondisi aktual dibandingkan dengan prediksi sistem untuk memperoleh TP, TN, FP, dan FN

**Tabel 8.** Pengujian Aktual Video dan CCTV

| No | Sumber Video | Kondisi Aktual | Prediksi Sistem | Hasil |
|---|---|---|---|---|
| 1 | Video Kecelakaan 1 | Kecelakaan | Kecelakaan | TP |
| 2 | Video Kecelakaan 2 | Kecelakaan | Kecelakaan | TP |
| 3 | Video Kecelakaan 3 | Kecelakaan | Tidak terdeteksi | FN |
| 4 | Video Kecelakaan 4 | Kecelakaan | Kecelakaan | TP |
| 5 | Video Kecelakaan 5 | Kecelakaan | Kecelakaan | TP |
| 6 | RTSP CCTV 1 | Non-kecelakaan | Non-kecelakaan | TN |
| 7 | RTSP CCTV 2 | Non-kecelakaan | Non-kecelakaan | TN |
| 8 | RTSP CCTV 3 | Non-kecelakaan | Kecelakaan | FP |
| 9 | RTSP CCTV 4 | Non-kecelakaan | Non-kecelakaan | TN |
| 10 | RTSP CCTV 5 | Non-kecelakaan | Non-kecelakaan | TN |

Berdasarkan Hasil Pengujian tersebut, sistem menghasilkan empat *True Positive*, empat *True Negative*, satu *False Positive*, dan satu *False Negative*. Nilai Precision dan Recall masing-masing sebesar 80%, sedangkan F1-Score sebesar 80% dan *False Positive Rate* sebesar 20%. Hasil tersebut menunjukkan bahwa masih terdapat kemungkinan kesalahan deteksi ketika sistem diterapkan pada video nyata, sehingga performa mekanisme deteksi kecelakaan tidak cukup dinilai hanya dari mAP@50 kelas kecelakaan

**Tabel 9.** *Confusion Matrix* Deteksi Kecelakaan

| Kondisi Aktual | Prediksi Kecelakaan | Prediksi Non-Kecelakaan |
|---|---|---|
| Kecelakaan | 4 (TP) | 1 (FN) |
| Non-kecelakaan | 1 (FP) | 4 (TN) |

**Tabel 10.** Metrik Evaluasi Deteksi Kecelakaan

| Metrik | Hasil |
|---|---|
| TP | 4 |
| TN | 4 |
| FP | 1 |
| FN | 1 |
| Accuracy | 80% |
| Precision | 80% |
| Recall | 80% |
| F1-Score | 80% |
| False Positive Rate | 20% |

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

Berdasarkan hasil pengujian tersebut, sistem menghasilkan empat *True Positive*, empat *True Negative*, satu *False Positive*, dan satu *False Negative*. Nilai Precision dan Recall masing-masing sebesar 80%, sedangkan F1-Score sebesar 80% dan *False Positive Rate* sebesar 20%. Hasil tersebut menunjukkan bahwa masih terdapat kemungkinan kesalahan deteksi ketika sistem diterapkan pada video nyata, sehingga performa mekanisme deteksi kecelakaan tidak cukup dinilai hanya dari mAP@50 kelas kecelakaan.

**Tabel 11.** *Confusion Matrix* Hasil Pengujian Model YOLOv8n pada Data Testing

| Predicted \ True | Kecelakaan | Bus | Mobil | Motor | Truk | BG |
|---|---|---|---|---|---|---|
| Kecelakaan | 39 | 0 | 4 | 1 | 0 | 18 |
| Bus | 0 | 26 | 2 | 0 | 0 | 5 |
| Mobil | 2 | 1 | 182 | 0 | 1 | 55 |
| Motor | 0 | 0 | 1 | 146 | 0 | 59 |
| Truk | 1 | 9 | 9 | 0 | 60 | 15 |
| BG | 11 | 0 | 10 | 19 | 9 | 0 |

Berdasarkan hasil *testing*, sebagian besar objek berhasil diklasifikasikan pada kelas yang sesuai, dengan 39 objek kecelakaan terdeteksi benar, 14 objek kecelakaan diklasifikasikan ke kelas lain atau *background*, dan 23 prediksi kecelakaan berasal dari objek non-kecelakaan; kesalahan klasifikasi terutama terjadi pada kelas mobil, motor, dan truk, seperti 56 objek motor yang diprediksi sebagai mobil, yang dapat dipengaruhi oleh kemiripan bentuk, ukuran objek, jarak kamera, dan kedekatan antar kendaraan. Hasil ini menunjukkan bahwa mAP@50 kelas kecelakaan sebesar 0,881 belum cukup untuk menggambarkan performa deteksi kecelakaan pada video, sehingga evaluasi dilengkapi dengan pengujian khusus pada video kecelakaan dan *live streaming* CCTV menggunakan TP, TN, FP, FN, Precision, Recall, F1-Score, dan *False Positive Rate* untuk menilai performa pada tingkat video secara terpisah.

**Pembahasan Perbandingan dan Keterbatasan**

Hasil penelitian menunjukkan bahwa model YOLOv8n yang digunakan memperoleh nilai mAP@50 sebesar 0,8623 (86,23%), sedangkan sistem mampu memproses video pada rentang 25–30 FPS. Perkasa et al. (2024) menerapkan YOLOv8 untuk *vehicle counting* dan melaporkan nilai mAP sebesar 82,77% pada hasil evaluasi model. Dengan demikian, nilai mAP@50 penelitian ini lebih tinggi dibandingkan nilai mAP yang dilaporkan Perkasa et al. (2024). Namun, perbandingan tersebut perlu dipahami secara hati-hati karena terdapat perbedaan dataset, jumlah kelas, konfigurasi model, skenario pengujian, dan definisi metrik yang digunakan.

Amsyah et al. (2025) mengembangkan sistem *vehicle counting* berbasis YOLOv8m dan melakukan validasi menggunakan *ground truth* hasil penghitungan manual. Pada sumber *live CCTV*, penelitian tersebut menghasilkan Precision 98,96%, Recall 96,59%, dan F1-Score 97,74% pada arah *upstream*, sedangkan pada arah *downstream* diperoleh Precision 100%, Recall 95,64%, dan F1-Score 97,73%. Pada video rekaman berkualitas tinggi, metrik yang dilaporkan mencapai 100%. Penelitian Amsyah et al. (2025) menunjukkan bahwa kualitas dan stabilitas sumber video berpengaruh terhadap performa deteksi, terutama ketika sistem diterapkan pada *live streaming*. Berbeda dengan penelitian tersebut yang berfokus pada *vehicle counting*, penelitian ini mengintegrasikan deteksi kecelakaan, *tracking*, *vehicle counting*, estimasi kecepatan, dan notifikasi kecelakaan dalam satu sistem berbasis web.

Pada aspek *tracking*, DeepSORT digunakan untuk mempertahankan identitas kendaraan melalui *Unique ID*. Identitas tersebut selanjutnya digunakan untuk mendukung *vehicle counting* dan estimasi kecepatan. Pada penelitian Perkasa et al. (2024), penghitungan kendaraan dilakukan menggunakan YOLOv8 dan OpenCV, sedangkan penelitian tersebut juga mencatat bahwa kualitas *frame rate* dan *frame drop* dapat memengaruhi akurasi sistem serta merekomendasikan pengujian *real-time* dan penggunaan identitas unik kendaraan. Penelitian ini memperluas fungsi tersebut dengan menggunakan DeepSORT sehingga kendaraan dapat dipantau berdasarkan *Tracking ID* dan digunakan sebagai dasar untuk beberapa proses analisis secara bersamaan.

Pada estimasi kecepatan, hasil pengujian langsung menunjukkan bahwa ketepatan estimasi dipengaruhi oleh jarak kendaraan terhadap kamera. Pada posisi dekat kamera, *error* terhadap *speedometer* sebesar 3,33%, meningkat menjadi 12,50% pada jarak menengah, dan mencapai 28,89% ketika kendaraan berada jauh dari kamera. Hasil ini menunjukkan bahwa performa estimasi kecepatan tidak hanya bergantung pada model deteksi dan *tracking*, tetapi juga dipengaruhi oleh perspektif kamera, ukuran objek pada citra, dan besarnya perpindahan objek antar-*frame*. Dengan demikian, penelitian ini memberikan informasi tambahan yang belum menjadi fokus utama penelitian Perkasa et al. (2024) maupun Amsyah et al. (2025), yaitu hubungan antara posisi kendaraan terhadap kamera dan ketepatan estimasi kecepatannya.

Beberapa kondisi dapat menyebabkan penurunan performa sistem. *Occlusion* dapat menyebabkan kendaraan kehilangan *Tracking ID*, sedangkan kepadatan lalu lintas tinggi dapat membuat

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

*bounding box* saling berdekatan atau bertumpang tindih. Pencahayaan rendah dan hujan dapat menurunkan kualitas fitur visual, sementara sudut kamera dan jarak objek memengaruhi perspektif serta estimasi kecepatan. Objek berukuran kecil juga lebih sulit dideteksi karena informasi visual yang tersedia lebih terbatas. Kondisi tersebut sejalan dengan temuan penelitian terdahulu bahwa kualitas video dan stabilitas *stream* dapat memengaruhi performa deteksi pada penerapan CCTV secara langsung.

Sistem juga menyediakan pengaturan *confidence threshold* melalui antarmuka web sehingga pengguna dapat menyesuaikan sensitivitas deteksi berdasarkan karakteristik sumber video. Pengaturan tersebut dapat membantu menyaring prediksi dengan tingkat keyakinan rendah, tetapi tidak menghilangkan kemungkinan *false positive* dan *false negative*. Oleh karena itu, hasil penelitian menunjukkan bahwa integrasi fungsi analisis dalam satu *pipeline* memberikan cakupan yang lebih luas, tetapi performa setiap komponen tetap dipengaruhi oleh karakteristik data dan kondisi lingkungan pengamatan.

## KESIMPULAN

Berdasarkan hasil implementasi dan evaluasi yang telah dilakukan, penelitian ini berhasil mengembangkan sistem analisis lalu lintas berbasis YOLOv8n dan DeepSORT yang mengintegrasikan deteksi kendaraan, pelacakan objek, estimasi kecepatan, penghitungan jumlah kendaraan, dan deteksi kecelakaan ke dalam satu aplikasi berbasis web. Kontribusi utama penelitian terletak pada integrasi seluruh proses analisis tersebut dalam satu *pipeline* sehingga informasi berupa *Unique ID*, estimasi kecepatan, *Current Vehicle*, *Total Vehicle*, *Average Speed*, serta notifikasi kecelakaan dapat disajikan secara *real-time* melalui satu dashboard.

Kinerja sistem dievaluasi melalui pelatihan 36 kombinasi *hyperparameter*, pengujian fungsional, pengujian *real-time*, validasi estimasi kecepatan, serta evaluasi khusus deteksi kecelakaan. Model terbaik menggunakan optimizer SGD dengan *epoch* 50, *batch size* 32, dan *learning rate* 0,001, menghasilkan Precision sebesar 0,8306, Recall sebesar 0,7633, F1-Score sebesar 0,7955, mAP@50 sebesar 0,8623, mAP@50–95 sebesar 0,517, dan mAP@50 kelas kecelakaan sebesar 0,8810. Pengujian sistem menghasilkan FPS 25–30 pada berbagai sumber video, sedangkan validasi estimasi kecepatan terhadap *speedometer* menunjukkan *error* 3,33% pada objek dekat kamera, 12,50% pada jarak menengah, dan 28,89% pada objek yang jauh. Evaluasi khusus deteksi kecelakaan pada lima video kecelakaan dan lima *live streaming* CCTV menghasilkan Precision 80%, Recall 80%, F1-Score 80%, dan *False Positive Rate* 20%. Hasil tersebut menunjukkan bahwa sistem mampu menyediakan analisis lalu lintas terpadu secara *real-time*, dengan performa yang tetap dipengaruhi oleh kondisi sumber video, jarak objek, kepadatan lalu lintas, dan karakteristik lingkungan pengamatan.

## DAFTAR PUSTAKA

Adhisatria, M. I. B. (2025). Perancangan Kontrol Adaptif Kepadatan Arus Lalu Lintas dengan Metode Convolutional Neural Network. *CYCLOTRON, 8*(01). https://doi.org/10.30651/cl.v8i01.24989

Amsyah, D. O., Harahap, L. S., & Fuady, A. F. (2025). Perancangan dan Analisis Kinerja Sistem Penghitung Lalu Lintas Otomatis Berbasis YOLOv8.

Badan Pusat Statistik. (2024). *Statistik Transportasi Indonesia 2024*. https://www.bps.go.id

Cai, B., Feng, Y., Wang, X., & Quddus, M. (2024). Highly Accurate Deep Learning Models for Estimating Traffic Characteristics from Video Data. *Applied Sciences, 14*(19), 8664. https://doi.org/10.3390/app14198664

Wang, C., & Liang, G. (2023, November). Overview of research on object detection based on YOLO. In *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering* (pp. 257-265). https://doi.org/10.1145/3652628.3652671

Ghahremannezhad, H., Shi, H., & Liu, C. (2022, June). Real-time accident detection in traffic surveillance using deep learning. In *2022 IEEE international conference on imaging systems and techniques* (IST) (pp. 1-6). IEEE. https://doi.org/10.1109/IST55454.2022.9827736

Husein, A. M., Lubis, K. N., Sidabutar, D. S., Yuanda, Y., & Waren, A. (2024). Computer vision-based intelligent traffic surveillance: multi-vehicle tracking and detection. *Sinkron: Jurnal Penelitian Teknik Informatika, 8*(1), 384-391. https://doi.org/10.33395/sinkron.v9i1.13204

Ikhsal, M. F., Dermawan, B. A., & Adam, R. I. (2023). Peningkatan Deteksi Kecelakaan di Jalan Raya Menggunakan Real-ESRGAN pada Citra CCTV Persimpangan Jalan. *Journal of Applied Informatics and Computing, 7*(1), 57-62.

Nimma, D., Al-Omari, O., Pradhan, R., et al. (2025). Object detection in real-time video surveillance using attention based transformer-YOLOv8 model. *Alexandria Engineering Journal, 118*, 482–495. https://doi.org/10.1016/j.aej.2025.01.032

Nurdin, A., Aji, B. S., Sudianto, Y., & Rafrin, M. (2023). YOLOv4 DeepSORT ANN for Traffic Collision Detection. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI),*

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

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

*12*(3), 338–351. https://doi.org/10.23887/janapati.v12i3.62923

Pei, R., Sholva, Y., & Novriando, H. (2023). Aplikasi Pelaporan Kecelakaan Lalu Lintas Menggunakan Metode Location Based Service Berbasis Progressive Web App. *JURISTI (Jurnal Riset Sains dan Teknologi Informatika), 1*(1), 56–68. https://doi.org/10.26418/juristi.v1i1.61440

Perkasa, W., Putra, W. P. N., Pradana, A. I., & Nurchim, N. (2024). Implementasi Sistem Penghitungan Volume Kendaraan Menggunakan Yolov8. *Jurnal Fasilkom, 14*(2), 443-450. https://doi.org/10.37859/jf.v14i2.7395

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