

Prediksi Kemacetan Lalu Lintas Menggunakan Data Citra CCTV Dan NLP Media Sosial

^{1*}Rahmat Nanda Eka Saputra, ²Risa Helilintar, ³Rony Heri Irawan

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

E-mail: ¹rnandajr758@gmail.com, ²risa.helilintar@gmail.com, ³rony@gmail.com

Penulis Korespondensi : Rahmat Nanda Eka Saputra

Abstrak— Kemacetan lalu lintas merupakan salah satu permasalahan utama di perkotaan yang dapat menyebabkan kerugian waktu, peningkatan konsumsi bahan bakar, dan menurunkan efisiensi mobilitas masyarakat. Penelitian ini bertujuan untuk mengembangkan sistem prediksi kemacetan lalu lintas menggunakan data citra CCTV dan media sosial berbasis *deep learning*. Metode yang digunakan adalah algoritma YOLOv8 untuk mendeteksi dan menghitung kendaraan secara real-time dari video CCTV, serta metode *Natural Language Processing* (NLP) untuk menganalisis informasi kondisi lalu lintas dari media sosial. Tahapan penelitian meliputi pengumpulan data, *preprocessing*, pelatihan model, dan evaluasi sistem. Evaluasi model dilakukan menggunakan metrik *precision*, *recall*, *F1-score*, dan *mean Average Precision (mAP)* untuk mengukur performa deteksi kendaraan dan klasifikasi informasi lalu lintas. Hasil penelitian menunjukkan bahwa integrasi data visual dan tekstual mampu memberikan informasi kondisi lalu lintas yang lebih lengkap, cepat, dan akurat dibandingkan penggunaan satu sumber data saja. Dengan demikian, sistem yang dikembangkan berpotensi mendukung implementasi *Intelligent Transportation System* dalam pemantauan dan prediksi kemacetan lalu lintas secara *real-time*.

Kata Kunci— YOLOv8, CCTV, NLP, Kemacetan Lalu Lintas, *Deep Learning*.

Abstract— Traffic congestion is one of the major urban problems that can cause time loss, increased fuel consumption, and reduced mobility efficiency. This study aims to develop a traffic congestion prediction system using CCTV image data and social media data based on deep learning. The methods used are the YOLOv8 algorithm for real-time vehicle detection and counting from CCTV videos, and Natural Language Processing (NLP) for analyzing traffic condition information from social media. The research stages include data collection, preprocessing, model training, and system evaluation. Model evaluation was carried out using precision, recall, F1-score, and mean Average Precision (mAP) metrics to measure the performance of vehicle detection and traffic information classification. The results show that the integration of visual and textual data can provide more complete, faster, and more accurate traffic condition information compared to using only a single data source. Therefore, the developed system has the potential to support the implementation of Intelligent Transportation Systems for real-time traffic monitoring and congestion prediction.

Keywords— YOLOv8, CCTV, NLP, Traffic Congestion, *Deep Learning*.

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



I. PENDAHULUAN

Kemacetan lalu lintas menjadi permasalahan utama di berbagai kota besar di Indonesia, seperti Medan, Malang, dan Palembang. Beberapa penelitian memanfaatkan teknologi *deep learning*

berbasis YOLO untuk mendeteksi kendaraan dan memantau kondisi lalu lintas secara otomatis melalui CCTV. Penelitian di Kota Medan menggunakan YOLOv4 dan *Euclidean Distance Tracker* untuk mendeteksi serta melacak kendaraan dengan akurasi yang meningkat hingga 100% pada sampel tertentu[1]. Sementara itu, penelitian di Kota Malang menggunakan YOLOv8 untuk mendeteksi kepadatan kendaraan dengan akurasi sebesar 78,07%[2], sedangkan penelitian di Palembang menerapkan YOLOv8 untuk mendeteksi pelanggaran *Yellow Box Junction* secara otomatis[3].

Selain digunakan untuk mendeteksi kemacetan, YOLO juga diterapkan dalam analisis kinerja lalu lintas dan sistem pengendalian lampu lalu lintas adaptif[4]. Penelitian yang mengintegrasikan YOLOv8 dengan standar PKJI 2023 menunjukkan hasil yang sangat baik dengan nilai mAP 0,948 dan akurasi perhitungan kendaraan sebesar 96%[5]. Penelitian lain menggunakan YOLOv4 untuk mendeteksi volume kendaraan dari CCTV dan menyesuaikan durasi lampu lalu lintas secara otomatis, sehingga mampu meningkatkan efisiensi lalu lintas dan mengurangi kemacetan[6]. Selain itu, simulasi lampu lalu lintas adaptif berbasis YOLOv4 juga menunjukkan performa yang baik dari sisi akurasi dan kecepatan deteksi secara *real-time*[7].

Beberapa penelitian juga mengembangkan sistem pendeteksian dan klasifikasi kendaraan berbasis kecerdasan buatan untuk mendukung sistem transportasi cerdas. Penggunaan YOLO dan CFNN terbukti mampu melakukan deteksi dan klasifikasi kendaraan secara *real-time* dengan akurasi tinggi[4]. Penelitian lain memanfaatkan YOLOv8 untuk menghitung kendaraan secara otomatis dan memperoleh akurasi hingga 89%, sehingga dinilai mampu menggantikan proses perhitungan manual yang kurang efisien[8]. Selain itu, terdapat penelitian deteksi marka jalan berbasis CNN dari video dashcam yang menunjukkan performa baik pada kondisi ideal, meskipun masih memiliki keterbatasan pada kondisi jalan kompleks dan belum sepenuhnya memenuhi standar *real-time*[9].

Tidak hanya data visual dari CCTV, beberapa penelitian juga memanfaatkan data teks dari media sosial untuk mendukung analisis kondisi lalu lintas. Penelitian berbasis Twitter menggunakan metode *deep learning* seperti CNN dan LSTM dengan *word embedding fastText* untuk melakukan klasifikasi multi-label kondisi lalu lintas, seperti kemacetan, kecelakaan, dan cuaca. Hasil terbaik diperoleh dari kombinasi LSTM dan *fastText* dengan akurasi hingga 0,95[10]. Selain itu, kajian literatur mengenai pemanfaatan kecerdasan buatan dalam manajemen lalu lintas menunjukkan bahwa teknologi AI, khususnya YOLOv5, mampu mengolah data secara *real-time* dengan tingkat akurasi tinggi hingga 98% dan banyak diterapkan melalui CCTV[11]. Penelitian ini menunjukkan bahwa YOLOv8 mampu mendeteksi dan menganalisis kemacetan lalu lintas secara *real-time* dengan tingkat akurasi yang tinggi[12]. Selain itu, mengembangkan sistem deteksi marka jalan berbasis CNN yang bekerja baik pada kondisi jalan lurus, namun belum optimal untuk kondisi kompleks dan *real-time*[13]. Penelitian ini menunjukkan bahwa YOLOv8 lebih unggul dalam deteksi kendaraan *real-time* dibandingkan SSD dengan akurasi dan *recall* yang tinggi[14]. Perkembangan teknologi CCTV memungkinkan proses pengawasan dilakukan secara lebih efektif melalui penyimpanan data digital, akses jarak jauh melalui internet, dan integrasi dengan berbagai teknologi pendukung lainnya[15].

Berdasarkan berbagai penelitian tersebut, dapat disimpulkan bahwa penerapan teknologi *deep learning*, khususnya algoritma YOLO, sangat efektif dalam mendeteksi kendaraan, menganalisis kepadatan lalu lintas, serta mendukung sistem transportasi cerdas secara *real-time*. Integrasi data visual dari CCTV dengan metode kecerdasan buatan mampu meningkatkan efisiensi pemantauan lalu lintas dan membantu mengurangi kemacetan. Selain itu, pemanfaatan data media sosial melalui metode NLP juga menunjukkan potensi besar dalam memberikan informasi kondisi lalu

lintas secara cepat dan akurat, sehingga kombinasi antara *computer vision* dan NLP dapat menjadi solusi yang lebih optimal dalam prediksi kemacetan lalu lintas.

II. METODE

Bagian ini menjelaskan tahapan penelitian yang meliputi pengumpulan dataset, preprocessing, pelatihan, hingga evaluasi model. Berikut penjelasannya :

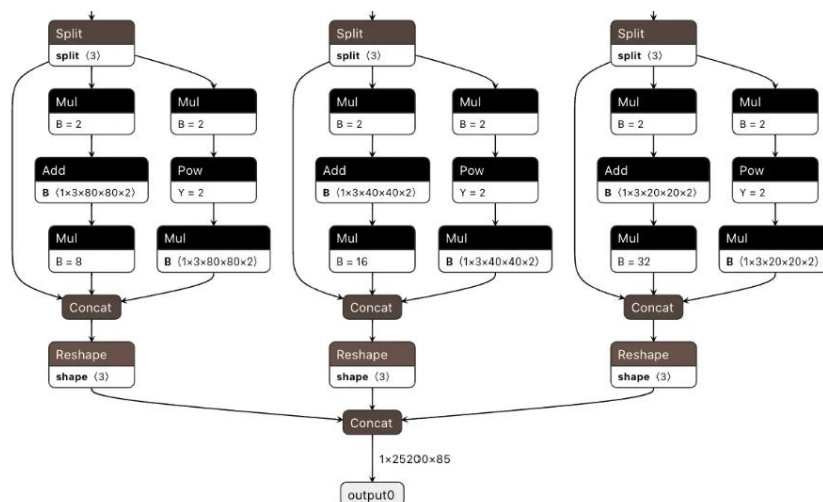
1. Input Gambar

Pada penelitian “*Simulasi Pengontrol Lampu Lalu Lintas Berbasis Deteksi Kendaraan dengan YOLOv4*”, data input yang digunakan berupa video rekaman CCTV lalu lintas dari ATCS DISHUB Kabupaten Bandung Barat. Video diambil pada beberapa kondisi waktu, yaitu pagi, siang, sore, dan malam untuk memperoleh variasi kondisi kepadatan lalu lintas dan pencahayaan. Rekaman video memiliki resolusi 360p dengan *frame rate* 30 fps sehingga dapat digunakan untuk mendeteksi kendaraan secara *real-time*. Kendaraan yang menjadi objek penelitian meliputi mobil, sepeda motor, bus, dan truk. Data video ini digunakan sebagai dasar analisis untuk menghitung jumlah kendaraan, memantau arus lalu lintas, serta mengatur simulasi lampu lalu lintas adaptif.

2. Preprocessing Data

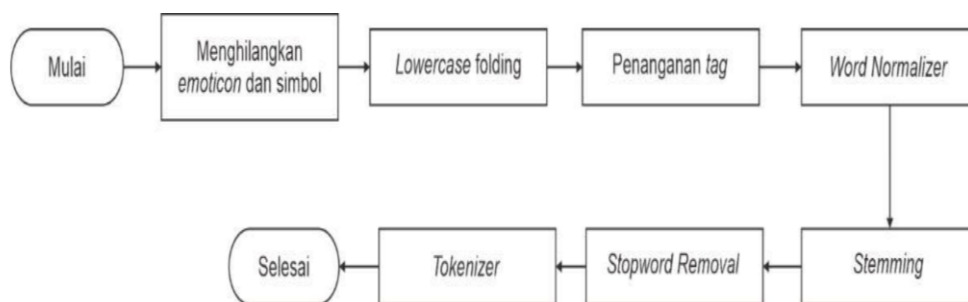
Tahap preprocessing dilakukan untuk menyiapkan data sebelum digunakan dalam proses pelatihan model YOLOv4. Pada tahap ini dilakukan proses anotasi citra kendaraan menggunakan *bounding box* untuk menandai posisi objek kendaraan pada setiap frame video. Setiap objek diberikan label sesuai kelas kendaraan, seperti mobil, motor, bus, dan truk. Hasil anotasi disimpan dalam format file *.txt* yang berisi informasi kelas objek, koordinat posisi, tinggi, dan lebar objek. Selain itu, dataset dibagi menjadi tiga bagian yaitu data training, *validation*, dan *testing* agar model dapat dilatih dan diuji secara objektif. Tahap *preprocessing* ini bertujuan agar model mampu mengenali objek kendaraan dengan lebih akurat pada berbagai kondisi lalu lintas.

3. Arsitektur Model



Gambar 1. Arsitektur Model YoloV8

YoloV8 merupakan algoritma *object detection* berbasis *Deep Learning* dan *Convolutional Neural Network* (CNN) yang dikembangkan oleh *Ultralytics* untuk mendeteksi objek secara *realtime* pada gambar maupun video. Arsitektur YoloV8 terdiri dari tiga bagian utama yaitu *backbone*, *neck*, dan *head*, dimana *backbone* berfungsi mengekstraksi fitur gambar, *neck* menggabungkan fitur dari berbagai skala, dan *head* menghasilkan prediksi berupa *bounding box*, *confidence score*, serta klasifikasi objek. Dibandingkan metode *object detection* lainnya, YoloV8 memiliki keunggulan dalam kecepatan dan akurasi karena mampu melakukan klasifikasi dan lokalisasi objek dalam satu kali proses inferensi. Pada penelitian ini, YoloV8 digunakan untuk mendeteksi kendaraan seperti mobil, motor, bus, dan truk pada video CCTV secara *realtime* sehingga dapat mendukung proses penghitungan kendaraan dan analisis kepadatan lalu lintas.



Gambar 2. Arsitektur NLP

Selain proses deteksi objek menggunakan YOLOv8, penelitian ini juga menerapkan tahapan *preprocessing* data teks sebagaimana ditunjukkan pada Gambar X. Proses diawali dengan menghilangkan *emoticon* dan simbol untuk membersihkan data dari karakter yang tidak diperlukan. Selanjutnya dilakukan *lowercase folding* untuk mengubah seluruh huruf menjadi huruf kecil agar data lebih seragam. Tahap berikutnya adalah penanganan *tag* dan *word normalization* untuk menghapus tag tertentu serta menormalisasi kata-kata yang tidak baku menjadi bentuk standar. Setelah itu dilakukan *stemming* untuk mengubah kata menjadi bentuk dasarnya, diikuti *stopword removal* guna menghilangkan kata-kata umum yang tidak memiliki makna penting dalam analisis. Tahap terakhir adalah *tokenizer*, yaitu proses memecah teks menjadi token atau kata-kata tunggal yang siap digunakan pada proses analisis lebih lanjut. Rangkaian tahapan ini bertujuan meningkatkan kualitas data sehingga dapat menghasilkan performa pemrosesan yang lebih optimal.

4. Training Model

Proses pelatihan model dilakukan menggunakan bahasa pemrograman Python dengan dukungan library *TensorFlow* dan *OpenCV*. Model YOLOv4 menggunakan *pretrained network* berbasis dataset *ImageNet* untuk mempercepat proses pembelajaran fitur objek kendaraan. Parameter pelatihan yang digunakan meliputi *batch size* 128, *mini-batch size* 32, *learning rate* 0.1, dan *momentum* 0.9. Selama proses training, model dilatih untuk mengenali berbagai jenis kendaraan seperti mobil, motor, bus, dan truk dari frame video CCTV. Selain itu, sistem juga memanfaatkan metode *tracking* untuk memantau pergerakan kendaraan yang telah terdeteksi sehingga penghitungan kendaraan dapat dilakukan secara otomatis dan *real-time*.

5. Evaluasi Model

Tahap evaluasi dilakukan menggunakan metode *confusion matrix* dan pengujian langsung pada video CCTV lalu lintas. Model diuji untuk mengetahui kemampuan dalam mendeteksi, melacak, dan menghitung kendaraan pada berbagai kondisi kepadatan lalu lintas. Pengujian dilakukan pada beberapa waktu berbeda, seperti pagi, siang, dan sore hari, guna melihat performa sistem pada kondisi lalu lintas yang bervariasi. Hasil evaluasi menunjukkan bahwa YOLOv4 mampu mendeteksi kendaraan secara *real-time* dengan performa yang baik dari sisi akurasi dan kecepatan. Sistem juga berhasil menghitung jumlah kendaraan berdasarkan jenisnya dan memberikan informasi kepadatan lalu lintas yang dapat dimanfaatkan dalam pengaturan lampu lalu lintas adaptif. Berikut beberapa rumusnya:

a. Intersection over Union (IoU)

IoU mengukur tumpang tindih antara *bounding box* prediksi ($(A_{\{p\}})$) dan *bounding box* target/ground truth ($(A_{\{g\}})$).

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{A_p \cap A_g}{A_p \cup A_g}$$

$$IoU = \frac{8000}{10000} = 0,8$$

b. Precision (Presisi)

Mengukur seberapa akurat model dalam memprediksi objek positif dari seluruh prediksi positif yang dibuat.

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

$$Precision = \frac{180}{180 + 20}$$

$$Precision = \frac{180}{200}$$

$$Precision = 0,90 = 90\%$$

TP (*True Positive*): Deteksi benar.

FP (*False Positive*): Deteksi salah (objek tidak ada tapi diprediksi ada).

c. Recall (Sensitivitas)

Mengukur kemampuan model dalam menemukan semua objek positif yang sebenarnya ada dalam gambar.

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

$$Recall = \frac{180}{180 + 30}$$

$$Recall = \frac{180}{210}$$

$$Recall = 0,857 = 85,7\%$$

FN (False Negative): Objek gagal terdeteksi (tidak terdeteksi).

d. *F1-Score*

Metrik gabungan yang mengombinasikan *precision* dan *recall* untuk memberikan keseimbangan antara keduanya.

$$F1 = 2 \times \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 = 2 \times \frac{0,90 \cdot 0,857}{0,90 + 0,857}$$

$$F1 = 2 \times \frac{0,7713}{1,757}$$

$$F1 = 0,878 = 87,8\%$$

e. *Mean Average Precision (mAP)*

Metrik utama dalam YOLOv8, yang mengevaluasi performa model pada berbagai ambang batas (*threshold*) IoU (biasanya 0.5 atau 0.5 hingga 0.95).

mAP50: mAP yang dihitung dengan IoU *threshold* 0.50.

mAP50-95: Rata-rata mAP pada berbagai tingkat *threshold* IoU mulai dari 0.50 hingga 0.95 (dengan langkah 0.05).

$$AP = \int_0^1 p(r)dr \Rightarrow mAP = \left(\frac{1}{N}\right) \sum_{i=1}^N AP_i$$

$$mAP = \frac{0,92 + 0,88 + 0,95 + 0,86 + 0,84}{5}$$

$$mAP = \frac{4,45}{5}$$

$$mAP = 0,89 = 89\%$$

N: Jumlah kelas.

p(r): Presisi sebagai fungsi dari *recall*.

f. *Loss Function (YOLOv8)*

Selama pelatihan, YOLOv8 meminimalkan total kerugian (*total loss*) yang terdiri dari tiga komponen:

$$L_{total} = L_{box} + L_{cls} + L_{obj}$$

$$L_{total} = 0,42 + 0,31 + 0,27$$

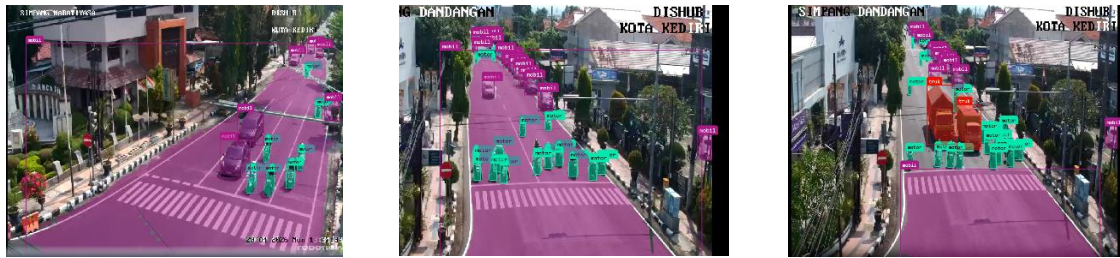
$$L_{total} = 1,00$$

III. HASIL DAN PEMBAHASAN

1. Input Gambar

Pada tahap ini, sistem menerima input berupa frame gambar dari video CCTV lalu lintas yang menampilkan berbagai kondisi kendaraan di jalan. Data citra dikelompokkan berdasarkan kategori objek kendaraan, seperti mobil, motor, bus, dan truk, serta kondisi kepadatan lalu lintas untuk mendukung proses analisis kemacetan. Seluruh data disusun dalam format yang

seragam dan diberikan label menggunakan *bounding box* agar proses pelatihan dan evaluasi model YOLOv8 dapat dilakukan secara terstruktur. Pada tahap pengujian, dataset digunakan untuk mengamati kemampuan model dalam mendeteksi dan menghitung kendaraan secara *real-time* sehingga hasil prediksi kemacetan dapat dianalisis dengan lebih akurat.



Lancar

Sedang

Macet

Gambar 3. Dataset

2. Preprocessing Data

a. Pengujian Akurasi Deteksi

Pengujian akurasi dilakukan untuk mengetahui kemampuan model YOLOv8 dalam mendeteksi dan mengklasifikasikan kendaraan secara tepat. Proses pengujian dilakukan menggunakan dataset pengujian yang berbeda dari dataset pelatihan. Setiap hasil deteksi dibandingkan dengan label *ground truth* untuk mengetahui tingkat kesesuaian prediksi model. Hasil pengujian menunjukkan bahwa model YOLOv8 mampu mendeteksi kendaraan dengan tingkat akurasi yang baik pada kondisi jalan normal dan pencahayaan cukup.

Tabel 1. Pengujian Akurasi

RINGKASAN AKURASI FINAL	
Model	: YOLOv8N Custom
Dataset	: 27 gambar (21 train / 6 val)
Epoch	: 100
mAP@0.5	: 46.33%
mAP@0.5 : 95	: 39.59%
Precision	: 60.99%
Recall	: 45.89%

b. Precision, Recall, dan mAP

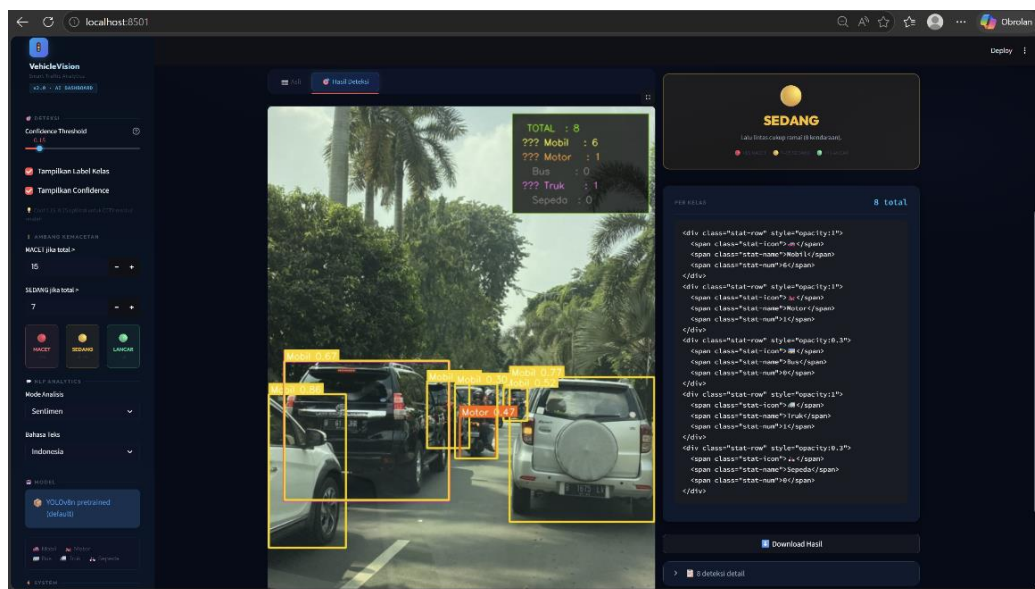
Pengukuran performa model dilakukan menggunakan metrik evaluasi object detection berupa *precision*, *recall*, dan *mean Average Precision* (mAP). *Precision* digunakan untuk mengukur tingkat ketepatan prediksi model terhadap objek yang terdeteksi. *Recall* digunakan untuk mengukur kemampuan model dalam menemukan seluruh objek kendaraan pada citra. Sedangkan mAP digunakan sebagai indikator utama performa *object detection* karena menggabungkan nilai *precision* dan *recall* pada berbagai *threshold*. Hasil evaluasi menunjukkan bahwa model YOLOv8 memiliki performa yang cukup baik dalam mendeteksi kendaraan pada video CCTV lalu lintas.

Tabel 2. Precision, Recall, dan mAP

Metrik	Penjelasan	Target
mAP@0.5	Mean Average Precision pada IoU 0.5 – metrik utama	≥ 70%
mAP@0.5 : 95	mAP rata-rata IoU 0.5-0.95 – lebih ketat	≥ 50%
Precision	Dari semua deteksi, berapa % yang benar	≥ 75%
Recall	Dari semua kendaraan nyata, berapa % terdeteksi	≥ 70%

c. Pengujian pada Kondisi CCTV

Gambar dibawah ini menampilkan antarmuka sistem deteksi kendaraan berbasis YOLOv8 yang memanfaatkan rekaman CCTV untuk mendeteksi dan mengklasifikasikan kendaraan secara otomatis. Hasil deteksi ditunjukkan melalui *bounding box* yang mengelilingi objek kendaraan, lengkap dengan label kelas dan nilai *confidence score*. Selain itu, sistem juga menampilkan informasi jumlah kendaraan berdasarkan kategori yang terdeteksi, seperti mobil dan motor, serta total kendaraan secara keseluruhan. Berdasarkan hasil pengujian, sistem berhasil mendeteksi delapan kendaraan pada area pengamatan, sehingga dapat digunakan untuk memantau kondisi lalu lintas dan mendukung analisis tingkat kepadatan kendaraan secara *real-time*.

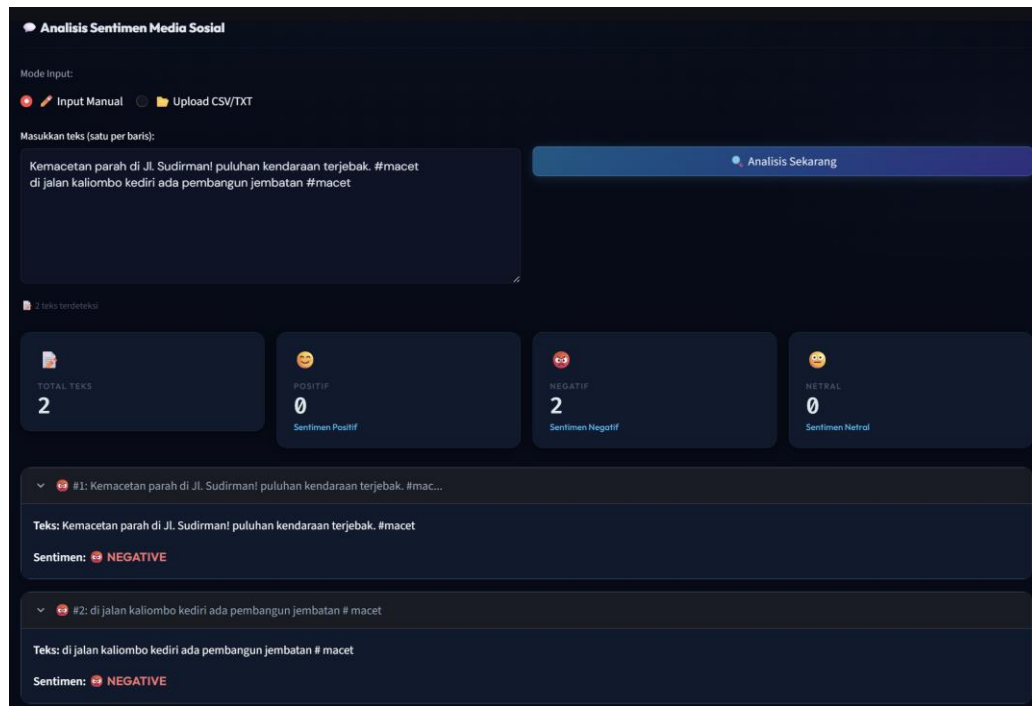


Gambar 4. Pengujian Pada Kondisi CCTV

d. Pengujian Teks dari Sosial Media

Gambar kedua menampilkan antarmuka sistem analisis sentimen media sosial yang digunakan untuk mengidentifikasi opini masyarakat terkait kondisi lalu lintas berdasarkan teks yang dimasukkan secara manual atau melalui file CSV/TXT. Sistem mengklasifikasikan setiap teks ke dalam kategori sentimen positif, negatif, atau netral. Pada contoh pengujian, dua teks yang membahas kemacetan lalu lintas berhasil diklasifikasikan sebagai sentimen *negatif*. Hasil analisis disajikan dalam bentuk statistik yang menunjukkan jumlah total teks dan distribusi sentimennya, sehingga dapat

digunakan untuk memahami persepsi masyarakat terhadap kondisi lalu lintas serta mendukung pengambilan keputusan dalam pengelolaan transportasi.



Gambar 5. Teks dari Media Sosial

3. Arsitektur Model

Penelitian ini menggunakan arsitektur *deep learning* yang mengintegrasikan YOLOv8 untuk pengolahan citra CCTV dan LSTM untuk analisis teks media sosial guna mendukung prediksi kemacetan lalu lintas secara *real-time*. YOLOv8 berperan dalam mendeteksi dan mengklasifikasikan kendaraan melalui komponen *backbone*, *neck*, dan *head* yang menghasilkan informasi lokasi objek, jenis kendaraan, serta tingkat kepercayaan deteksi. Sementara itu, LSTM digunakan untuk menganalisis informasi lalu lintas dari media sosial dengan memahami hubungan antar kata dalam teks yang telah melalui tahap *preprocessing* dan *word embedding*. Integrasi kedua metode tersebut memungkinkan sistem memanfaatkan data visual dan tekstual secara bersamaan sehingga dapat menghasilkan prediksi kondisi lalu lintas yang lebih cepat, akurat, dan adaptif.

4. Training Model

Proses training dilakukan untuk melatih model YOLOv8 dan LSTM agar mampu mendeteksi kendaraan dari data CCTV serta menganalisis informasi lalu lintas dari media sosial secara akurat. YOLOv8 dilatih menggunakan dataset citra kendaraan yang telah dianotasi dengan berbagai parameter pelatihan, seperti *epoch*, *batch size*, *learning rate*, dan pretrained model, serta dievaluasi menggunakan metrik *loss*, *precision*, *recall*, dan mAP. Sementara itu, model LSTM dilatih menggunakan data teks media sosial yang telah melalui tahap *preprocessing* dan *word embedding* untuk mengklasifikasikan kondisi lalu lintas berdasarkan konteks informasi yang disampaikan. Melalui proses pelatihan ini, sistem diharapkan mampu mengintegrasikan data visual dan tekstual sehingga dapat menghasilkan prediksi kemacetan lalu lintas yang lebih akurat dan *real-time*.

5. Evaluasi Model

Tahap evaluasi dilakukan untuk mengukur performa sistem dalam mendeteksi kendaraan dari CCTV dan menganalisis informasi lalu lintas dari media sosial menggunakan data uji yang tidak terlibat dalam proses pelatihan. Model YOLOv8 dievaluasi menggunakan metrik *precision*, *recall*, *F1-score*, *mean Average Precision (mAP)*, dan *confusion matrix* untuk menilai kemampuan deteksi serta klasifikasi kendaraan, sedangkan model LSTM dievaluasi menggunakan akurasi, *precision*, *recall*, dan *F1-score* untuk mengukur keberhasilan klasifikasi informasi lalu lintas dari teks media sosial. Hasil evaluasi kedua model kemudian dianalisis untuk mengetahui efektivitas integrasi data visual dan tekstual dalam mendukung prediksi kemacetan lalu lintas secara *real-time*, sehingga sistem dapat memberikan informasi kondisi lalu lintas yang lebih cepat, akurat, dan konsisten.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma YOLOv8 pada data citra CCTV serta metode NLP pada media sosial mampu mendukung sistem prediksi kemacetan lalu lintas secara efektif dan *real-time*. YOLOv8 berhasil mendeteksi serta menghitung kendaraan dengan akurasi yang baik, sedangkan NLP mampu menganalisis informasi kondisi lalu lintas dari media sosial seperti kemacetan, kecelakaan, dan gangguan jalan. Integrasi data visual dan tekstual memberikan informasi lalu lintas yang lebih lengkap, sehingga sistem dapat meningkatkan proses pemantauan, analisis, dan prediksi kemacetan secara lebih cepat, akurat, dan adaptif terhadap kondisi lalu lintas perkotaan.

DAFTAR PUSTAKA

- [1] J. Manurung, N. Azizi, D. Anastasya, N. Valentino, A. Sanjaya, and K. Saputra, "Deteksi Kemacetan dengan Deep Learning YOLOv4 dan Euclidean Distance Tracker pada Jalan Raya di Kota Medan," *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, vol. 8, no. 1, pp. 57–63, Feb. 2023, doi: 10.32528/justindo.v8i1.220.
- [2] A. Surya and I. Wahyuni, "Deteksi Kepadatan Lalu Lintas Pada CCTV Publik Pemerintah Kota Malang Menggunakan YOLOv8," *SPECTA Journal of Technology*, vol. 9, no. 2, pp. 136–149, Aug. 2025, doi: 10.35718/specta.v9i2.8481367.
- [3] A. Kholik and M. D. Fauzi, "BULLETIN OF COMPUTER SCIENCE RESEARCH Deteksi Pelanggaran Durasi Berhenti Kendaraan pada Area Yellow Box Junction Menggunakan Algoritma YOLOv8," *Media Online*, vol. 6, no. 3, pp. 870–876, 2026, doi: 10.47065/bulletincsr.v6i3.1055.
- [4] M. A. Putri, Riyono, and I. Setiawan Wibisono, "SmartTraffic-CNN: Deteksi dan Estimasi Jumlah Kendaraan Secara Otomatis Menggunakan Deep Learning dan Ekstraksi Fitur," *Jurnal Algoritma*, vol. 22, no. 2, Nov. 2025, doi: 10.33364/algoritma/v.22-2.2943.
- [5] A. Ahdan Amir, A. Delviana Lihara, M. Nabil, A. Info Kata Kunci, and K. Lalu Lintas, "Sistem Analisis Kinerja Lalu Lintas Berbasis Deep Learning dengan Arsitektur You Only Look Once (YOLO)," vol. 6, no. 1, pp. 179–190, 2026, doi: 10.51454/decode.v6i1.1503.
- [6] F. Raihan Wahidin, W. Witanti, E. Ramadhan, U. Jenderal Achmad Yani, K. Bandung, and P. Korespondensi, "PENGONTROLAN LAMPU LALU LINTAS MENGGUNAKAN TEKNOLOGI DETEKSI KENDARAAN YOLOV4," vol. 12, no. 5, pp. 2355–7699, 2025.

- [7] R. Ramadan, H. Setiawan, and U. Jenderal Achmad Yani, “Studi Kasus pada Video Rekaman CCTV.”
- [8] N. J. Hayati, D. Singasatia, M. R. Muttaqin, T. Informatika, S. Tinggi, and T. Wastukencana, “OBJECT TRACKING MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO)v8 UNTUK MENGHITUNG KENDARAAN,” *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023, [Online]. Available: <https://universe.roboflow.com/>
- [9] A. R. Muttakin and D. Haryanto, “Sistem Penghitungan Kendaraan Otomatis Menggunakan Metode Object Detection Algoritma CNN”, [Online]. Available: <https://journal.uniku.ac.id/index.php/buffer>
- [10] L. Atikah, D. Purwitasari, and N. Suciati, “DETEKSI KEJADIAN LALU LINTAS PADA TEKS TWITTER DENGAN PENDEKATAN KLASIFIKASI MULTI-LABEL BERBASIS DEEP LEARNING”, doi: 10.25126/jtiik.202295206.
- [11] M. Hasan Fikri, M. Miftahul Ulum, and N. Latifah Dwi Mutiara Sari, “Literature review Implementasi kecerdasan buatan untuk Prediksi dan Pengelolaan Kemacetan Lalu Lintas di Perkotaan,” *Seminar Nasional Informatika-FTI UPGRIS*, vol. 2, 2024.
- [12] R. Hidayattullah, N. Suarna, I. Ali, and D. I. Efendi, “DEEP LEARNING ALGORITMA YOLOV8 UNTUK MENINGKATKAN ANALISIS KEPADATAN LALU LINTAS,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 2, Apr. 2025, doi: 10.23960/jitet.v13i2.5749.
- [13] F. Maulana Ridho, A. Fahrudi Setiawan, and J. Dedy Irawan, “IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK DETEKSI MARKA JALAN,” 2025.
- [14] 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, doi: 10.52436/1.jutif.2025.6.1.3916.
- [15] R. H. Irawan *et al.*, [i]. 2019. [Online]. Available: www.penerbitlitnus.co.id