



**ANALISA DETEKSI KENDARAAN BERDASARKAN JENIS
DAN KECEPATAN MENGGUNAKAN KAMERA CCTV
BERBASIS ALGORITMA YOLO V11 DI LINGKUNGAN
UNIVERSITAS JEMBER**

*diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana pada
program studi Teknik Elektro*

SKRIPSI

Oleh

**Putra Gama Supriyadi
211910201085**

**KEMENTERIAN PENDIDIKAN TINGGI SAINS DAN TEKNOLOGI
UNIVERSITAS JEMBER
FAKULTAS TEKNIK
S1 TEKNIK ELEKTRO
JEMBER
2025**

PERSEMBAHAN

Segala puji dan syukur penulis panjatkan kepada Allah SWT atas karunia kesehatan, rahmat, serta hidayah-Nya yang telah memberi kesempatan kepada penulis untuk menyelesaikan skripsi ini sebagai bagian dari syarat meraih gelar Sarjana. Meskipun hasilnya belum sempurna, penulis merasa bersyukur dan bangga telah mencapai tahap ini, di mana skripsi ini dapat diselesaikan tepat waktu. Dengan penuh rasa syukur, karya ini penulis dedikasikan untuk:

1. Bapak Supiyadi, Ibu Mujayasih, Kakak Putri Elsyia Supriyadi, terimakasih banyak karena telah memberikan dukungan baik dukungan materi, moral dan doa untuk kelancaran penulis untuk mendapatkan gelar sarjana.
2. Bapak Ir. Khairul Anam, S.T., M.T., Ph.D., IPU., ASEAN Eng. selaku Dosen Pembimbing Utama yang telah memberi bimbingan dan dukungan sehingga dapat menyelesaikan skripsi ini dengan baik.
3. Bapak Andrita Ceriana Eska, S.T., M.T. selaku Dosen Pembimbing Anggota yang telah telah memberi masukan dan dukungan sehingga dapat menyelesaikan skripsi ini dengan baik.
4. Dosen Teknik Elektro Universitas Jember terimakasih karena telah memberi ilmunya dan dengan sabar mengajari penulis selama berkuliah di Universitas Jember.
5. TOWER'21 teman seangkatan yang telah berjuang bersama dan telah membantu baik selama perkuliahan maupun selama mengerjakan skripsi saya.
6. Pihak UPT TIK terimakasih karena telah bersedia memberikan akses pada IP CCTV guna menyelesaikan penelitian ini.

MOTTO

“The best of humans are those who are most beneficial to others.”

(HR. Ahmad)

“Hargai setiap momen, sebab baik maupun buruk, semuanya akan berlalu.”

(Penulis)

PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini :

Nama : Putra Gama Supriyadi

NIM : 211910201085

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul: *Analisa Deteksi Kendaraan Berdasarkan Jenis Dan Kecepatan Menggunakan Kamera CCTV Berbasis Algoritma Yolo V11 Di Lingkungan Universitas Jember* adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 18 Juli 2025

Yang menyatakan,

Putra Gama Supriyadi

NIM 211910201085

HALAMAN PERSETUJUAN

Skripsi berjudul *Analisa Deteksi Kendaraan Berdasarkan Jenis Dan Kecepatan Menggunakan Kamera CCTV Berbasis Algoritma Yolo V11 Di Lingkungan Universitas Jember* telah diuji dan disetujui oleh Fakultas Teknik Universitas Jember pada:

Hari : Jumat

Tanggal : 18 Juli 2025

Tempat : Fakultas Teknik Universitas Jember

| Pembimbing | Tanda Tangan |
|---|--------------|
| 1. Pembimbing Utama Nama : Ir. Khairul Anam, S.T., M.T., Ph.D., IPU., ASEAN Eng. NIP : 197804052005011002 | (.....) |
| 2. Pembimbing Anggota Nama : Andrita Ceriana Eska, S.T., M.T. NIP : 198706172023211021 | (.....) |
| Penguji | |
| 1. Penguji Utama Nama : Ir. Gamma Aditya Rahardi, S.T, M.T. NIP : 199101282019031018 | (.....) |
| 2. Penguji Anggota Nama : Ali Rizal Chaidir, S.T., M.T. NIP : 198907082023211024 | (.....) |

ABSTRAK

Penelitian ini bertujuan untuk menganalisis kemampuan algoritma YOLO V11 dalam mendeteksi jenis dan kecepatan kendaraan dari rekaman CCTV di Universitas Jember. Permasalahan utama adalah bagaimana akurasi dan kecepatan sistem bekerja secara real-time. Tujuan lainnya adalah meningkatkan efektivitas pemantauan lalu lintas dengan sistem otomatis berbasis deep learning. Metodologi penelitian ini menggunakan pendekatan eksperimental dengan menerapkan algoritma YOLO V11 untuk mendeteksi kendaraan. Data diambil dari rekaman CCTV yang dikonversi menjadi frame, kemudian dilabeli dan dilatih menggunakan YOLO. Pengujian dilakukan secara offline dan real-time dengan evaluasi melalui metrik mAP, precision, recall, serta validasi kecepatan kendaraan menggunakan garis deteksi. Hasil penelitian menunjukkan bahwa algoritma YOLO V11 mampu mendeteksi jenis dan kecepatan kendaraan dengan akurasi tinggi. Model terbaik mencapai nilai mAP50 sebesar 0,99274 dan F1-score sebesar 0,98. Sistem juga berhasil menampilkan kecepatan kendaraan secara real-time dengan tingkat error rendah, rata-rata 3,18% untuk motor dan 2,72% untuk mobil. Kesimpulan dari penelitian ini menunjukkan bahwa YOLO V11 efektif dalam mendeteksi jenis dan kecepatan kendaraan secara real-time dengan akurasi tinggi. Sistem ini dapat diimplementasikan untuk mendukung pemantauan lalu lintas. Disarankan pengembangan lebih lanjut mencakup peningkatan akurasi di kondisi pencahayaan rendah dan perluasan deteksi untuk lebih banyak jenis kendaraan.

Kata kunci: YOLO V11, Deteksi Kendaraan, Kecepatan Kendaraan, CCTV, *Deep Learning*

ABSTRACT

This study aims to analyze the ability of the YOLO V11 algorithm in detecting vehicle types and speeds from CCTV footage at the University of Jember. The main problem is how the accuracy and speed of the system work in real-time. Another goal is to improve the effectiveness of traffic monitoring with an automated system based on deep learning. The methodology of this study uses an experimental approach by applying the YOLO V11 algorithm to detect vehicles. Data is taken from CCTV footage that is converted into frames, then labeled and trained using YOLO. Testing is carried out offline and in real-time with evaluation through mAP, precision, recall metrics, and validation of vehicle speed using detection lines. The results show that the YOLO V11 algorithm is able to detect vehicle types and speeds with high accuracy. The best model achieved an mAP50 value of 0.99274 and an F1-score of 0.98. The system also managed to display vehicle speeds in real-time with a low error rate, an average of 3.18% for motorbikes and 2.72% for cars. The conclusion of this study shows that YOLO V11 is effective in detecting vehicle types and speeds in real-time with high accuracy. This system can be implemented to support traffic monitoring. Further development is suggested including improving accuracy in low light conditions and expanding detection to more vehicle types.

Keywords: YOLO V11, Vehicle Detection, Vehicle Speed, CCTV, Deep Learning

RINGKASAN

Penerapan algoritma YOLOv11 pada sistem pendeteksi kendaraan difokuskan untuk mengidentifikasi jenis kendaraan serta mengukur kecepatan secara otomatis berbasis rekaman CCTV. Model dilatih menggunakan data citra kendaraan yang diperoleh dari lingkungan Universitas Jember, dengan proses anotasi dua kelas utama: mobil dan motor. Data tersebut diolah menggunakan teknik *deep learning* berbasis *Convolutional Neural Network* (CNN), dengan konfigurasi *training* yang mencakup *epoch*, *batch size*, *optimizer*, serta augmentasi ringan untuk meningkatkan generalisasi model.

Pengujian dilakukan terhadap tiga varian model YOLOv11, yaitu YOLOv11n, YOLOv11s, dan YOLOv11m. Hasilnya menunjukkan bahwa YOLOv11m memiliki akurasi tertinggi (mAP50 sebesar 0,993), namun dengan waktu inferensi lebih lambat. Sementara itu, YOLOv11n menunjukkan performa mendekati akurat dengan waktu inferensi tercepat di antara ketiga varian. Evaluasi metrik lain seperti *precision*, *recall*, dan *F1-score* mendukung keandalan model dalam klasifikasi objek kendaraan.

Dari perbandingan antara hasil deteksi YOLOv11 dengan data pengukuran kecepatan aktual dari alat *Speed Gun*, diperoleh rata-rata error kecepatan sebesar 3,18% untuk motor dan 2,72% untuk mobil. Pengujian real-time dengan input kamera CCTV juga menunjukkan hasil yang cukup akurat, dengan integrasi sistem ke dalam modul LED Matrix untuk visualisasi kecepatan secara langsung. Temuan ini menegaskan bahwa penggunaan algoritma YOLOv11 pada sistem pemantauan lalu lintas mampu memberikan hasil deteksi yang cepat, akurat, dan dapat diandalkan untuk aplikasi di lapangan secara *real-time*.

PRAKATA

Assalamualaikum warahmatullahi wabarokatuh.

Alhamdulillahirabbil'alaamiin.

Penulis mengucapkan syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “*Analisa Deteksi Kendaraan Berdasarkan Jenis Dan Kecepatan Menggunakan Kamera Cctv Berbasis Algoritma YOLO V11 Di Lingkungan Universitas Jember.*” Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada bidang Elektronika dan Sistem Kendali, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Penulis juga menyampaikan terima kasih kepada semua pihak yang telah memberikan bantuan, dukungan, dan motivasi selama proses penyusunan hingga selesainya skripsi ini.

1. Bapak Supiyadi, Ibu Mujayasih, Kakak Putri Elsyia Supriyadi, terimakasih banyak karena telah memberikan dukungan baik dukungan materi, moral dan doa untuk kelancaran penulis untuk mendapatkan gelar sarjana.
2. Dosen Teknik Elektro Universitas Jember terimakasih karena telah memberi ilmunya dan dengan sabar mengajari penulis selama berkuliah di Universitas Jember.
3. TOWER'21 teman seangkatan yang telah berjuang bersama dan telah membantu baik selama perkuliahan maupun selama mengerjakan skripsi saya.

Penulis berharap skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang akademik, serta bermanfaat bagi para pembaca. Penulis juga terbuka dan menyambut baik kritik maupun saran yang bersifat membangun sebagai bahan evaluasi untuk penyempurnaan skripsi ini di masa mendatang. Kritik dan saran dapat disampaikan melalui alamat surel: putragamasupriyadi@gmail.com.

Penulis juga menyampaikan permohonan maaf apabila terdapat kekurangan maupun keterbatasan dalam penulisan skripsi ini.

DAFTAR ISI

| | |
|--|-------------|
| HALAMAN JUDUL | i |
| PERSEMBAHAN..... | ii |
| MOTTO | iii |
| PERNYATAAN ORISINALITAS..... | iv |
| HALAMAN PERSETUJUAN..... | v |
| ABSTRAK | vi |
| RINGKASAN | viii |
| PRAKATA..... | ix |
| DAFTAR ISI..... | x |
| DAFTAR TABEL..... | xii |
| DAFTAR GAMBAR..... | xiii |
| BAB 1. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Penelitian | 3 |
| 1.4 Tujuan Penelitian..... | 3 |
| 1.5 Manfaat Penelitian..... | 4 |
| BAB 2. TINJAUAN TEORI..... | 5 |
| 2.1 Kajian Literatur | 5 |
| 2.2 Kajian Pustaka..... | 7 |
| 2.2.1. NVIDIA Jetson Nano | 7 |
| 2.2.2. ESP32 | 8 |
| 2.2.3. Pengukuran Kecepatan Kendaraan..... | 8 |
| 2.2.4. IP CAM CCTV..... | 9 |
| 2.2.5. YOLO (<i>You Only Look Once</i>)..... | 10 |
| 2.3 Pengembangan Hipotesis | 14 |
| BAB 3. METODOLOGI PENELITIAN..... | 15 |
| 3.1 Lokasi dan Waktu Penelitian..... | 15 |
| 3.2 Sampel/Subyek Penelitian..... | 15 |
| 3.3 Alat/Instrumen Penelitian..... | 15 |
| 3.4 Prosedur / Tahapan Penelitian..... | 15 |
| 3.5 Diagram Sistem Keseluruhan..... | 16 |
| 3.6 Desain Perancangan Alat | 17 |
| 3.7 Data Pengujian | 18 |
| BAB 4. PEMBAHASAN | 20 |

| | | |
|-----------------------|---|-----------|
| 4.1 | Pengolahan Dataset dan Parameter <i>Training</i> | 20 |
| 4.2 | Hasil Evaluasi Parameter dan Pengujian YOLOv11 | 22 |
| 4.2.1 | Pengaruh Jumlah <i>Epoch</i> Terhadap Nilai Akurasi Model | 22 |
| 4.2.2 | Evaluasi <i>Batch Size</i> terhadap Akurasi Deteksi | 22 |
| 4.2.3 | Perbandingan <i>Optimizer</i> terhadap Konvergensi Model | 23 |
| 4.2.4 | Perbandingan Model YOLOv11 terhadap Akurasi dan Efisiensi | 24 |
| 4.3 | Pengujian Pendeteksian Jenis dan Jumlah Kendaraan | 34 |
| 4.4 | Pengujian Estimasi Kecepatan Kendaraan | 35 |
| 4.5 | Pengujian Tampilan <i>Running Text</i> | 37 |
| 4.6 | Pengujian Program pada Jetson Nano | 38 |
| 4.7 | Pengujian Sistem secara <i>Real Time</i> | 39 |
| BAB 5. | PENUTUP | 42 |
| 5.1 | Kesimpulan | 42 |
| 5.2 | Saran | 42 |
| DAFTAR PUSTAKA | | 43 |
| LAMPIRAN | | 45 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Literatur YOLO..... | 6 |
| Tabel 2.2 Karakteristik Yolo..... | 12 |
| Tabel 3.1 <i>Hardware</i> , <i>Software</i> , dan <i>Platform</i> | 15 |
| Tabel 4.1 Hasil pengujian nilai <i>Epoch</i> | 22 |
| Tabel 4.2 Hasil pengujian nilai <i>Batch</i> | 23 |
| Tabel 4.3 Hasil pengujian <i>optimizer</i> | 24 |
| Tabel 4.4 Hasil pengujian model YOLOV11 | 24 |
| Tabel 4.5 Hasil pengujian pendeteksian kendaraan | 35 |
| Tabel 4.6 Hasil pengujian kecepatan kendaraan | 36 |
| Tabel 4.7 Hasil pengujian <i>running text</i> | 37 |
| Tabel 4.8 Hasil Pengujian Pada Jetson Nano..... | 38 |
| Tabel 4.9 Hasil Pengujian <i>Real Time</i> | 40 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 NVIDIA Jetson Nano | 7 |
| Gambar 2.2 ESP32 | 8 |
| Gambar 2.3 Fisik IP <i>Camera CCTV</i> | 9 |
| Gambar 2.4 Arsitektur Jaringan YOLO | 10 |
| Gambar 3.1 Diagram Penelitian | 15 |
| Gambar 3.2 Diagram Sistem Keseluruhan | 16 |
| Gambar 3.3 Desain <i>Box Running Text</i> | 17 |
| Gambar 3.4 Schematic Rangkaian Alat | 18 |
| Gambar 4.1 <i>Labeling Dataset</i> pada <i>Roboflow</i> | 20 |
| Gambar 4.2 Hasil training menggunakan model YOLOV11n | 25 |
| Gambar 4.3 <i>Confusion matrix</i> | 25 |
| Gambar 4.4 Kurva nilai F1 terhadap nilai <i>Confidence</i> | 26 |
| Gambar 4.5 Kurva nilai <i>Precision</i> terhadap nilai <i>Confidence</i> | 26 |
| Gambar 4.6 Kurva nilai <i>Recall</i> terhadap nilai <i>Confidence</i> | 27 |
| Gambar 4.7 Kurva nilai <i>Precision</i> terhadap nilai <i>Recall</i> | 27 |
| Gambar 4.8 Hasil <i>training</i> menggunakan model YOLOV11s | 28 |
| Gambar 4.9 <i>Confusion matrix</i> | 28 |
| Gambar 4.10 Kurva nilai F1 terhadap nilai <i>Confidence</i> | 29 |
| Gambar 4.11 Kurva nilai <i>Precision</i> terhadap nilai <i>Confidence</i> | 29 |
| Gambar 4.12 Kurva nilai <i>Recall</i> terhadap nilai <i>Confidence</i> | 30 |
| Gambar 4.13 Kurva nilai <i>Precision</i> terhadap nilai <i>Recall</i> | 30 |
| Gambar 4.14 Hasil training menggunakan model YOLOV11m | 31 |
| Gambar 4.15 <i>Confusion matrix</i> | 31 |
| Gambar 4.16 Kurva nilai F1 terhadap nilai <i>Confidence</i> | 32 |
| Gambar 4.17 Kurva nilai <i>Precision</i> terhadap nilai <i>Confidence</i> | 32 |
| Gambar 4.18 Kurva nilai <i>Recall</i> terhadap nilai <i>Confidence</i> | 33 |
| Gambar 4.19 Kurva nilai <i>Precision</i> terhadap nilai <i>Recall</i> | 33 |
| Gambar 4.20 <i>Speed Gun</i> | 36 |
| Gambar 4.21 Tampilan <i>Website</i> pada <i>localhost</i> | 39 |
| Gambar 4.22 Grafik Hasil Pengujian Sistem secara <i>Real Time</i> | 41 |

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi telah mendorong transformasi dalam berbagai bidang, termasuk keselamatan dan investigasi kecelakaan lalu lintas. Dengan hadirnya kecerdasan buatan seperti *deep learning* dan deteksi objek *real-time*, analisis visual kini dapat dilakukan lebih cepat dan akurat. Salah satu teknologi unggulan adalah YOLO (*You Only Look Once*), yang memungkinkan pendeteksian objek secara langsung dari rekaman kamera pengawas. (Tejashree et al., 2025) menunjukkan bahwa algoritma YOLO sangat efektif dalam mendeteksi dan melacak kendaraan secara *real-time* dengan akurasi tinggi dan efisiensi waktu yang baik, bahkan dalam kondisi lalu lintas yang padat dan beragam. Lebih lanjut, penelitian dari (Naveenkumar et al., 2025) membuktikan bahwa penerapan YOLO dalam sistem deteksi kecelakaan otomatis mampu mengirimkan notifikasi *real-time* ke layanan darurat, sehingga mempercepat respons terhadap insiden di jalan raya. Penerapan teknologi ini tidak hanya meningkatkan efektivitas investigasi forensik, tetapi juga membuka peluang dalam pengembangan sistem peringatan dini dan pencegahan kecelakaan secara proaktif (Erik et al., 2025).

Perkembangan transportasi di Indonesia terus meningkat, dengan jumlah kendaraan mencapai 133,6 juta unit pada tahun 2019, yang berkontribusi pada masalah kemacetan. Para ahli dan peneliti berupaya mencari solusi melalui pencacahan jenis dan jumlah kendaraan untuk memahami karakteristik lalu lintas dan mencegah penumpukan kendaraan di lokasi tertentu. Namun, metode manual sering memakan waktu dan rentan terhadap kesalahan. Dengan kemajuan teknologi, sistem otomatis berbasis *deep learning* mulai digunakan untuk mendeteksi citra kendaraan, sehingga perhitungan jumlah dan jenis kendaraan menjadi lebih cepat dan akurat, menawarkan solusi yang lebih efisien dalam menangani masalah lalu lintas (Hayati et al., 2023). (Juliansyah et al., 2024) mengembangkan sistem monitoring kepadatan kendaraan berbasis YOLOv8 dan *OpenCV* yang mampu mengklasifikasikan kendaraan secara *real-time* dengan akurasi hingga 96%, menunjukkan potensi besar dalam mengurangi ketergantungan pada pemantauan

manual. Di sisi lain, sistem manajemen lalu lintas cerdas berbasis AI yang dikembangkan oleh (Kamble et al., 2025) juga menggunakan algoritma YOLO dan *OpenCV* untuk mengatur durasi lampu lalu lintas secara adaptif berdasarkan kepadatan kendaraan, menghasilkan efisiensi lalu lintas yang lebih tinggi tanpa memerlukan perubahan infrastruktur besar.

Seiring dengan pengembangan teknologi *artificial intelligence*, khususnya di bidang *computer vision*, memungkinkan sistem kamera pengawas untuk mendeteksi, menginterpretasikan, dan merespons informasi visual secara efektif. Kemajuan ini menawarkan solusi efisien bagi pemerintah daerah, perusahaan manajemen lalu lintas, pengembang properti, dan perusahaan iklan dalam memperoleh data yang akurat dan *real-time* untuk perencanaan lalu lintas, keamanan, serta analisis pemasaran. Salah satu solusi yang berkembang adalah penggunaan algoritma YOLO dalam sistem perhitungan jumlah kendaraan berdasarkan jenis kendaraan (Kusuma et al., 2024). Selain itu, sistem *SmartFlow-YOLOv11* yang dirancang oleh (Talaat et al., 2025) berhasil mengintegrasikan deteksi kendaraan secara *real-time* dan pengoptimalan arus lalu lintas, menghasilkan mAP sebesar 92,4% serta mampu menyesuaikan sinyal lalu lintas secara dinamis untuk meningkatkan efisiensi transportasi perkotaan.

Adapun penelitian sebelumnya yang digunakan sebagai referensi adalah penelitian H. Gibran, B. Purnama, G. Kosala. (2024). *Technomedia Journal (TMJ)* yang berjudul Pengoptimalan Pengukuran Kepadatan Jalan Raya Dengan Cctv Menggunakan Metode Yolov8 , yang membahas pengukuran kepadatan lalu lintas menggunakan CCTV dan metode YOLO v8. Penelitian ini berfokus pada deteksi kendaraan secara real-time dan menganalisis kinerja sistem menggunakan metrik seperti *precision*, *recall*, dan *F1 score*, dengan hasil yang lebih akurat pada siang hari dibanding malam hari. Penelitian lainnya oleh M. Faizin, 2023, mengimplementasikan YOLO v5 untuk mendeteksi jenis dan kecepatan kendaraan berdasarkan dataset dari CCTV, dengan pengujian parameter seperti *Epoch*, *Batch size*, dan *Optimizer*.

Merujuk pada penelitian sebelumnya, dalam penelitian ini penulis akan mengimplementasikan penggunaan YOLO dalam sistem deteksi dan pengelolaan

lalu lintas berbasis CCTV dengan mengambil judul " Analisa Deteksi Kendaraan Berdasarkan Jenis Dan Kecepatan Menggunakan Kamera CCTV Berbasis Algoritma YOLO v11 Di Lingkungan Universitas Jember " untuk mengembangkan penelitian sebelumnya.

1.2 Rumusan Masalah

Berdasarkan penjabaran latar belakang diatas maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana kemampuan penggunaan metode YOLO V11 dalam melakukan pendeteksian jenis kendaraan?
2. Bagaimana kemampuan penggunaan metode YOLO V11 dalam melakukan pengukuran kecepatan kendaraan?

1.3 Batasan Penelitian

Adapun batasan masalah pada penelitian tugas akhir ini agar tidak terlalu luas, diantaranya sebagai berikut:

1. Penelitian ini tidak membahas secara mendalam mengenai teknik pengolahan citra gambar, melainkan berfokus pada implementasi dan kinerja algoritma YOLO (*You Only Look Once*) dalam mendeteksi objek dari tangkapan video kamera CCTV.
2. Menggunakan NVIDIA Jetson Nano dan laptop sebagai alat pemrosesan.
3. Menggunakan lokasi CCTV arah masuk dari *Double Way* Universitas Jember
4. Posisi penempatan CCTV dalam kondisi pencahayaan tetap yaitu pagi, siang, dan sore hari dengan tampilan latar yang tidak mengalami perubahan.
5. Penelitian ini hanya mendeteksi dua kelas yaitu mobil dan motor.

1.4 Tujuan Penelitian

Merujuk pada latar belakang yang telah diuraikan sebelumnya, maka penelitian ini bertujuan untuk:

1. Mengetahui kemampuan penggunaan metode YOLO V11 dalam melakukan pendeteksian jenis kendaraan.

2. Mengetahui kemampuan penggunaan metode YOLO V11 dalam melakukan pengukuran kecepatan kendaraan.

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat dimanfaatkan sebagai alat bantu dalam meningkatkan efisiensi pemantauan lalu lintas di lingkungan Universitas Jember.

BAB 2. TINJAUAN TEORI

Bab kedua ini akan menguraikan berbagai teori yang mendukung penyelesaian penelitian berjudul “Analisa Deteksi Kendaraan Berdasarkan Jenis dan Kecepatan Menggunakan Kamera Cctv Berbasis Algoritma Yolo V11 di Lingkungan Universitas Jember”. Teori-teori yang dibahas mencakup cara kerja pemrosesan gambar (*image processing*), metode YOLO yang digunakan, serta elemen-elemen dalam citra (*image*). Pemrosesan gambar ini nantinya akan berperan dalam membantu mengukur kecepatan kendaraan dengan menggunakan rumus yang telah ditentukan.

2.1 Kajian Literatur

Penelitian (Faizin, 2023) menunjukkan bahwa algoritma YOLOv5 dapat mendeteksi jenis kendaraan dengan akurasi yang memadai pada siang hari, namun performa menurun pada malam hari karena tantangan pencahayaan rendah. Rata-rata error deteksi untuk mobil mencapai 17.39% pada malam hari, sementara untuk motor sebesar 8.19%. Pengukuran kecepatan juga menunjukkan rata-rata error sebesar 10.22% untuk mobil dan 4.45% untuk motor.

Pada penelitian yang dilakukan (Jauhari, 2024) mengimplementasikan YOLOv8 dengan nilai mAP50 sebesar 0.962 dan mAP50-90 sebesar 0.603. Sistem mendeteksi kendaraan dengan kecepatan pemrosesan rata-rata 16.6 ms per gambar, menggunakan dataset 324 gambar. Sistem ini menunjukkan akurasi tinggi pada deteksi kendaraan di siang hari, namun membutuhkan pengembangan lebih lanjut untuk meningkatkan performa pada kondisi malam hari.

Penelitian (Gibran et al., 2024) menunjukkan bahwa YOLOv8 dapat mengklasifikasikan kendaraan dengan baik pada siang hari (mAP 1.0), namun akurasi menurun pada malam hari (mAP 0.5) karena tantangan pencahayaan rendah. Fluktuasi pada grafik *loss* dan metrik evaluasi memperlihatkan bahwa sistem memerlukan lebih banyak *training* untuk kondisi malam.

Penelitian (Alif, 2024) menunjukkan bahwa YOLOv11 merupakan inovasi terbaru yang unggul dalam deteksi kendaraan, dengan mAP 0,743 secara

keseluruhan, melebihi YOLOv8 dan YOLOv10. Model ini mencapai presisi hingga 98,2% untuk motor dan 86,3% untuk bus, serta *recall* tinggi seperti 93% pada *threshold* rendah. Dengan kecepatan inferensi tinggi dan kemampuan bekerja di kondisi sulit, YOLOv11 sangat potensial untuk sistem transportasi cerdas, kendaraan otonom, dan pengawasan lalu lintas *real-time*.

Tabel 2.1 Literatur YOLO

| Penelitian | Judul | Jenis YOLO | Hasil Data |
|-----------------------|--|------------|--|
| (Faizin, 2023) | Implementasi Algoritma Yolo Pada Modul Kamera Untuk Deteksi Jenis Dan Kecepatan Kendaraan | YOLO v5 | <ul style="list-style-type: none"> - Akurasi memadai pada siang hari. - Rata-rata error deteksi mobil pada malam hari: 17.39%. - Rata-rata error deteksi motor: 8.19%. - Error pengukuran kecepatan mobil: 10.22%. - Error pengukuran kecepatan motor: 4.45%. |
| (Jauhari, 2024) | Sistem Deteksi Kendaraan Menggunakan Metode Yolo V8 Melalui Rekaman Cctv | YOLO v8 | <ul style="list-style-type: none"> - mAP50: 0.962. - mAP50-90: 0.603. - Kecepatan pemrosesan: 16.6 ms/gambar. - Dataset: 324 gambar. |
| (Gibran et al., 2024) | Pengoptimasian Pengukuran Kepadatan Jalan Raya Dengan Cctv Menggunakan Metode Yolov8 | YOLO v8 | <ul style="list-style-type: none"> - mAP siang hari: 1.0. - mAP malam hari: 0.5. |
| (Alif, 2024) | YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems | YOLO v11 | <ul style="list-style-type: none"> - mAP@0.5: 76,8% - mAP@0.75: 68,1% - mAP@0.5:0.95: 48,5% - Performa pada berbagai kondisi: Tinggi akurasi dan robustness di berbagai kondisi pengujian |

2.2 Kajian Pustaka

2.2.1. NVIDIA Jetson Nano

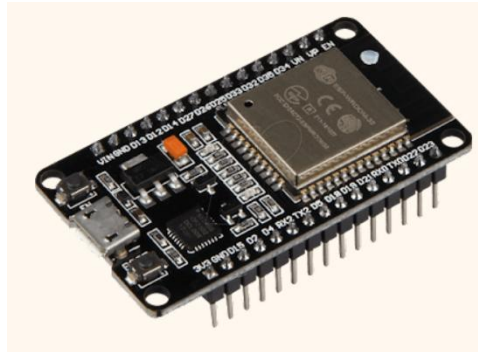


Gambar 2.1 NVIDIA Jetson Nano

(Sumber : <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/>)

NVIDIA Jetson Nano merupakan perangkat komputasi berbasis edge yang dirancang untuk menjalankan aplikasi kecerdasan buatan secara *real-time* dengan efisien dan berdaya rendah. Perangkat ini dibekali GPU Maxwell 128-core, CPU Quad-core ARM Cortex-A57, dan RAM 4 GB, serta mendukung berbagai *framework AI* seperti *TensorFlow*, *PyTorch*, dan *YOLO*. Jetson Nano juga dilengkapi dengan *slot microSD* untuk penyimpanan, *port USB 3.0*, *port Gigabit Ethernet*, dan antarmuka kamera melalui koneksi CSI. Dimensinya yang ringkas (69 mm × 45 mm) membuatnya ideal untuk implementasi sistem cerdas di lapangan, termasuk dalam pengolahan citra dari kamera CCTV tanpa bergantung pada koneksi internet, sebagaimana pendekatan *Dew Computing*. Dalam penelitian ini, Jetson Nano digunakan sebagai platform utama untuk mendeteksi jenis dan kecepatan kendaraan menggunakan algoritma YOLOv11, dengan keunggulan pengolahan data langsung di perangkat (*on-device inference*) yang mengurangi latensi dan memungkinkan pengawasan lalu lintas di lingkungan Universitas Jember secara efisien (Valladares et al., 2021).

2.2.2. ESP32



Gambar 2.2 ESP32

(Sumber : <https://ozami.co.id/wp-content/uploads/2024/09/Mengenal-ESP32-Mikrokontroler-IoT-1536x864.png>)

ESP32 adalah mikrokontroler 32-bit yang dikembangkan oleh Espressif Systems, sebagai penerus dari ESP8266. Mikrokontroler ini dilengkapi dengan modul Wi-Fi dan Bluetooth Low Energy (BLE) terintegrasi, sehingga sangat sesuai untuk berbagai aplikasi Internet of Things (IoT). Selain mendukung pemrograman menggunakan Arduino IDE, ESP32 juga memiliki memori cukup besar, lebih banyak pin input/output, dan fitur dual-core yang memungkinkan pemrosesan lebih cepat dan efisien. Dengan kemampuannya untuk menangani komunikasi nirkabel serta fleksibilitas pengembangan, ESP32 menjadi pilihan populer dalam proyek-proyek pemantauan, kontrol, dan otomasi berbasis IoT maupun AI (Pratama & Kiswantono, 2023).

2.2.3. Pengukuran Kecepatan Kendaraan

Pengukuran kecepatan kendaraan pada sistem pengolahan citra dilakukan dengan prinsip membandingkan jarak dan waktu tempuh suatu objek. Proses ini dimulai dengan mendeteksi koordinat titik tengah kendaraan dari frame video secara berurutan. Setelah kendaraan melintasi dua garis bantu yang telah ditentukan sebagai titik awal dan akhir pengukuran, waktu tempuh dihitung berdasarkan selisih waktu antar frame saat kendaraan melewati masing-masing garis. Kemudian, kecepatan dihitung dengan membagi jarak aktual antar garis bantu dengan waktu yang dibutuhkan kendaraan untuk melintasinya. Metode ini cukup efektif

digunakan pada sistem berbasis visi komputer karena memanfaatkan informasi spasial dan temporal secara langsung dari kamera (Susanto, 2020).

Persamaan menentukan kecepatan kendaraan dapat dituliskan sebagai berikut.

$$V = S/t \quad (2.1)$$

Dimana :

V = kecepatan yang akan dicari

S = jarak sebenarnya antara kedua garis bantu

t = waktu yang diperlukan objek untuk melewati kedua garis bantu

2.2.4. IP CAM CCTV



Gambar 2.3 Fisik IP Camera CCTV

(Sumber : <https://www.hikvision.com/id/products/IP-Products/Network-Cameras/>)

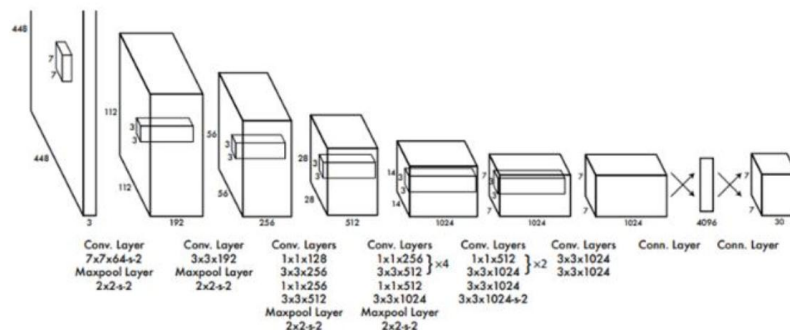
IP camera CCTV digunakan pertama kali pada tahun 1996. Istilah “IP Camera” atau “*Network Camera*” biasanya digunakan untuk sistem pengawasan keamanan. IP camera merupakan salah satu jenis kamera yang bisa digunakan untuk sistem pemantauan seperti pada toko, kantor, jalan raya serta dapat digunakan untuk mengirim dan menerima data melalui jaringan internet. Adapun keunggulan dari IP camera ini yaitu karena kinerjanya yang cukup kompleks sehingga penggunaan dan konfigurasinya menjadi lebih mudah (Susanto, 2020).

Dalam sistem pengawasan modern, kamera IP (*Internet Protocol*) CCTV memiliki peran penting untuk mendukung aplikasi deteksi objek berbasis algoritma *deep learning* seperti YOLO. Kamera IP dengan dukungan protokol RTSP (*Real-Time Streaming Protocol*) menjadi pilihan utama karena mampu menyediakan streaming video secara *real-time* untuk analisis otomatis. Beberapa jenis kamera yang kompatibel dengan YOLO, seperti Hikvision DS-2CD2043G0-I, Dahua IPC-

HFW1431S, Axis M2025-LE, Reolink RLC-410, dan Amcrest IP4M-1026B, memiliki karakteristik utama seperti resolusi tinggi (2 MP hingga 4 MP) untuk menghasilkan gambar yang jelas, dukungan RTSP untuk memudahkan integrasi, serta frame rate stabil yang memastikan kelancaran aliran video. Kombinasi karakteristik ini memungkinkan sistem deteksi berbasis YOLO bekerja untuk pemantauan keamanan, pengawasan lalu lintas (Sholahuddin et al., 2023).

2.2.5. YOLO (*You Only Look Once*)

You Only Look Once (YOLO) adalah algoritma berbasis *Convolutional Neural Network* (CNN) yang menggunakan pendekatan jaringan saraf tunggal untuk mendeteksi objek dalam gambar. Algoritma ini mampu memprediksi setiap *bounding box* dengan memanfaatkan fitur-fitur dari seluruh gambar secara langsung. Dalam satu evaluasi, YOLO dapat memprediksi *bounding box* serta probabilitas objek secara bersamaan. Arsitektur YOLO beroperasi dengan membagi gambar atau *frame* menjadi sejumlah bagian yang kemudian diberi *grid*, di mana setiap *grid* dianalisis untuk memprediksi probabilitas keberadaan objek pada tiap bagiannya, sehingga memungkinkan identifikasi objek secara efisien. Algoritma ini umumnya digunakan untuk mendeteksi berbagai jenis objek, seperti kendaraan, manusia, hewan, dan lain-lain, karena kemampuannya dalam mendeteksi objek secara *real-time* dengan tingkat akurasi yang tinggi (Muhlashin & Stefanie, 2023).



Gambar 2.4 Arsitektur Jaringan YOLO

(Muhlashin & Stefanie, 2023)

Algoritma YOLO beroperasi dengan empat pendekatan utama. Pertama, gambar dibagi menjadi *grid* $N \times N$, di mana setiap sel *grid* bertanggung jawab untuk

mendeteksi objek dan memprediksi kategori serta probabilitasnya. Kedua, YOLO menentukan *bounding box* yang mengelilingi objek dalam gambar dengan menggunakan representasi vektor

$$Y = [pc, bx, by, bh, bw, c1, c2] \quad (2.2)$$

yang menggambarkan efisiensi deteksi objek, koordinat pusat, ukuran *bounding box*, dan probabilitas kelas objek yang terdeteksi. Langkah ini sangat penting dalam *training* model untuk meningkatkan akurasi (Bamane et al., 2024).

Selanjutnya, *Intersection Over Union (IoU)* digunakan untuk mengukur tingkat tumpang tindih antara *bounding box* prediksi dan *ground truth*, yang membantu dalam meningkatkan akurasi lokasi objek. Terakhir, *Non-Maximum Suppression (NMS)* diterapkan untuk menghilangkan *bounding box* yang berlebihan atau tumpang tindih, hanya mempertahankan deteksi yang paling akurat dan relevan. Dengan demikian, YOLO dapat mendeteksi objek dengan cepat dan akurat, menjadikannya sangat efisien untuk aplikasi *real-time*. (Bamane et al., 2024). Berikut merupakan Tabel karakteristik YOLO dari versi 1 hingga versi 11.

Tabel 2.2 Karakteristik Yolo

| Versi YOLO | FPS | Akurasi (mAP) | Kelebihan | Kekurangan |
|------------|-------------------------------------|-------------------------------------|---|---|
| YOLOv1 | 45 FPS | mAP 63.4% | Cepat dengan arsitektur sederhana. | Tidak efisien mendeteksi objek kecil dan multi-skala. |
| YOLOv2 | 67 FPS | mAP 76.8% | Memperkenalkan anchor box, peningkatan akurasi dengan menggunakan Darknet. | Masih sulit mendeteksi objek kecil. |
| YOLOv3 | ~30 FPS | mAP 57.9% (COCO) | Multi-skala deteksi dengan tiga kepala deteksi. | Model lebih kompleks dan membutuhkan lebih banyak daya komputasi. |
| YOLOv4 | ~62 FPS | mAP 65.7% (COCO) | CSPDarknet-53 sebagai backbone, mendukung data augmentation dan peningkatan efisiensi pada inference. | Lebih lambat dibandingkan YOLO sebelumnya saat digunakan pada perangkat keras terbatas. |
| YOLOv5 | 140–150 FPS (GPU), 20–30 FPS (edge) | mAP@0.5: 55.3%, mAP@0.5:0.95: 34.2% | Mudah dilatih, ringan, dan cepat. Menggunakan CSPNet, PANet, residual structure, dan auto-anchor. Cocok untuk perangkat edge. | Deteksi objek kecil lebih rendah dibandingkan YOLOv6. Akurasi lebih rendah dibandingkan YOLOv7 untuk beberapa tugas spesifik. |
| YOLOv6 | 160–180 FPS (GPU), 25–35 FPS (edge) | | Peningkatan pada efisiensi inferensi dibandingkan YOLOv5. | Kurang didokumentasikan dalam eksperimen yang sama seperti YOLO lainnya. |

| Versi YOLO | FPS | Akurasi (mAP) | Kelebihan | Kekurangan |
|------------|------------------------------------|-------------------------------------|---|---|
| YOLOv7 | 23 FPS (Jetson Nano - YOLOv7-tiny) | mAP@0.5: 51.5%, mAP@0.5:0.95: 31.5% | Model lebih ringan dengan 35% lebih sedikit parameter dibandingkan YOLOv5. Menggunakan E-ELAN untuk efisiensi pembelajaran. | Memiliki tingkat false negative yang lebih tinggi dibandingkan YOLOv5 pada beberapa eksperimen. |
| YOLOv8 | ~70 FPS | AP50 84.4% | Menggunakan kepala decoupled, mendukung anchor-free deteksi, dan lebih ringan. | Keterbatasan dalam menangani gradien negatif pada dataset berkualitas rendah. |
| YOLOv9 | ~18–20 FPS (54 ms) | 0.935 | Akurasi tertinggi dalam deteksi karena PGI (Programmable Gradient Information) dan GELAN (Gradient Enhanced Lightweight Architecture Network) | Lambat dibandingkan YOLOv10 & v11, kebutuhan komputasi tinggi |
| YOLOv10 | ~52 FPS (19.3 ms) | 0.872 | NMS-Free inference, efisien, ringan, dual assignment head, cocok untuk real-time | Akurasi lebih rendah dibandingkan YOLOv9 & v11 |
| YOLOv11 | ~74 FPS (13.5 ms) | 0.921 | C3k2 block dan C2PSA module meningkatkan efisiensi, kecepatan, dan performa pada objek kecil & bertumpuk | Beberapa kelas akurasi sedikit di bawah YOLOv9, tetapi lebih seimbang |

2.3 Pengembangan Hipotesis

Hipotesis ini dibuat berdasarkan hasil penelitian dengan YOLOv11 yang akan digunakan oleh penulis. Diharapkan, penggunaan YOLOv11 dapat meningkatkan akurasi deteksi dan kecepatan pemrosesan.

Hipotesis:

1. Keakuratan Deteksi Jenis Kendaraan

Hipotesis Null (H_0): Metode YOLO V11 memiliki tingkat kemampuan yang rendah dalam mendeteksi jenis kendaraan.

Hipotesis Alternatif (H_1): Metode YOLO V11 memiliki tingkat kemampuan yang tinggi dalam mendeteksi jenis kendaraan.

2. Keakuratan Estimasi Kecepatan Kendaraan:

Hipotesis Null (H_0): Metode YOLO V11 memiliki tingkat kemampuan yang rendah dalam mengukur kecepatan kendaraan.

Hipotesis Alternatif (H_1): Metode YOLO V11 memiliki tingkat kemampuan yang tinggi dalam mengukur kecepatan kendaraan.

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tempat dan waktu penelitian, alat dan bahan, tahapan penelitian, diagram alir, perancangan alat, serta langkah-langkah pengambilan data.

3.1 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan dalam rentang waktu selama kurang lebih enam (6) bulan, yang mencakup pelaksanaan penelitian, perancangan, dan pengambilan data alat. Kegiatan penelitian ini dilakukan di Laboratorium CDAST (*Center for Development of Advance Science and Technology*) Universitas Jember.

3.2 Sampel/Subyek Penelitian

Sampel atau subjek penelitian yang digunakan dalam penelitian ini adalah video rekaman CCTV, yang kemudian diubah menjadi gambar dalam bentuk frame per frame. Gambar-gambar tersebut diberi label untuk dijadikan dataset yang akan digunakan dalam penerapan algoritma YOLO pada penelitian ini.

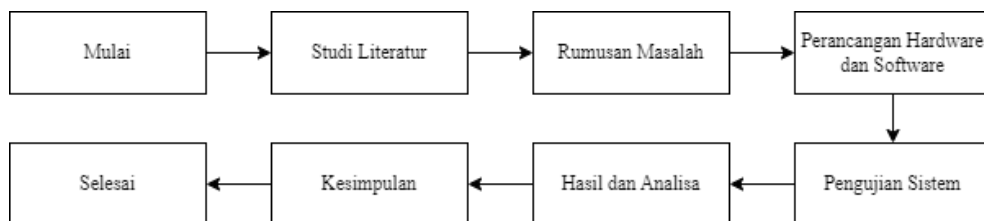
3.3 Alat/Instrumen Penelitian

Untuk Hardware, Software, dan Platform yang digunakan pada penelitian ini antara lain sebagai berikut:

Tabel 3.1 *Hardware , Software, dan Platform*

| Hardware | Software & Platform |
|--------------------|------------------------|
| Laptop | Python |
| NVIDIA Jetson Nano | Visual Studio Code |
| IP Cam CCTV | Video to JPG Converter |
| ESP32 | Roboflow |
| Led Matrix Max7219 | Kaggle Notebook |

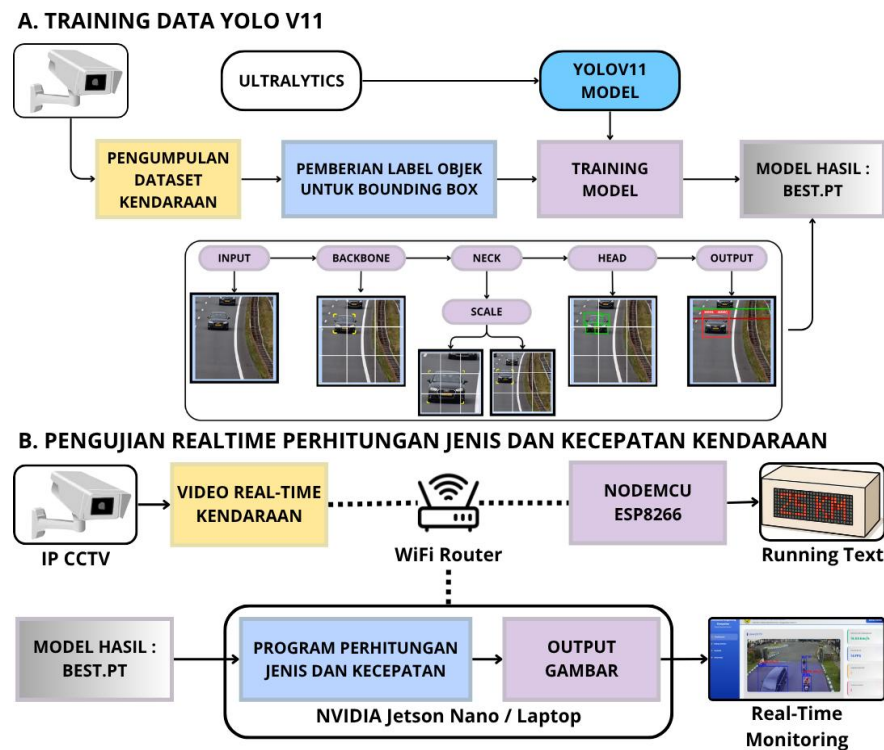
3.4 Prosedur / Tahapan Penelitian



Gambar 3.1 Diagram Penelitian

3.5 Diagram Sistem Keseluruhan

Supaya lebih mudah dalam memahami sistem kerja pada penelitian ini, maka dibuat desain sistem seperti pada gambar



Gambar 3.2 Diagram Sistem Keseluruhan

Diagram sistem keseluruhan terdiri dari dua tahap utama, yaitu tahap *Training* Data YOLOv11 dan tahap *Pengujian Realtime* Perhitungan Jenis dan Kecepatan Kendaraan.

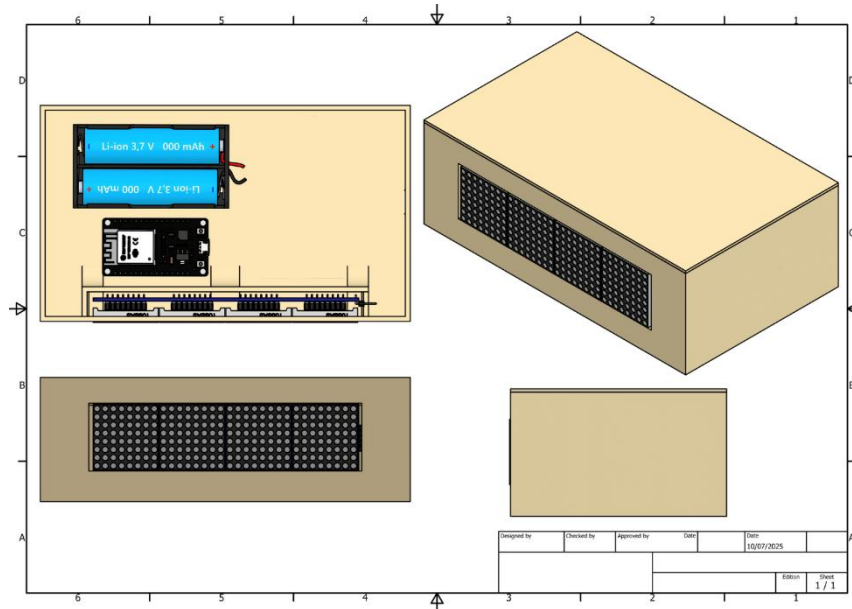
Pada tahap pertama, sistem diawali dengan proses pengumpulan dataset kendaraan yang direkam menggunakan kamera CCTV. Dataset ini kemudian diberi label berupa *bounding box* untuk menandai objek kendaraan seperti mobil dan motor. Setelah tahap pelabelan, dataset digunakan untuk proses *training* model menggunakan YOLOv11. Hasil dari proses *training* ini adalah model terlatih (*best.pt*) yang digunakan untuk mendeteksi kendaraan.

Tahap kedua adalah *pengujian real-time* untuk menghitung jenis dan kecepatan kendaraan. Sistem menerima input video *real-time* dari IP CCTV yang terhubung melalui jaringan WiFi. Video tersebut diproses oleh program perhitungan jenis dan kecepatan kendaraan yang dijalankan pada perangkat NVIDIA Jetson Nano atau

laptop. Program ini memanfaatkan model terlatih (*best.pt*) untuk mendeteksi jenis kendaraan serta menghitung kecepatannya berdasarkan waktu tempuh melewati dua garis bantu (ROI). Hasil deteksi jenis kendaraan dan kecepatan kemudian dikirimkan melalui modul ESP32 untuk ditampilkan di media *running text* LED MAX7219. Selain itu, output gambar deteksi juga ditampilkan dalam bentuk *monitoring real-time* sehingga pengguna dapat memantau langsung hasil deteksi kendaraan yang sedang melintas.

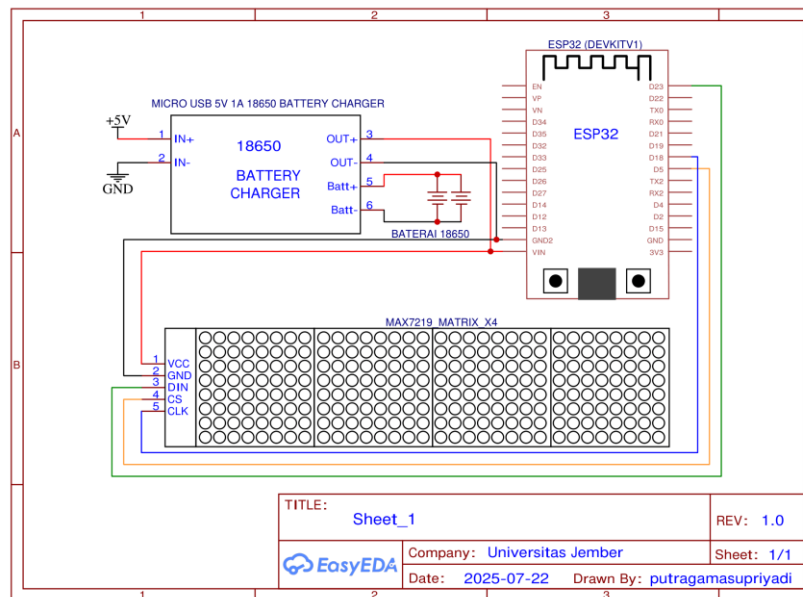
3.6 Desain Perancangan Alat

Desain alat dibuat dalam bentuk *box* yang dirancang untuk memuat seluruh komponen yaitu ESP32, LED matrix MAX7219, serta sumber daya berupa baterai.



Gambar 3.3 Desain *Box Running Text*

Box atau kotak ini didesain agar memiliki ukuran yang sesuai dengan dimensi LED matrix dan mikrokontroler, serta dilengkapi dengan lubang ventilasi untuk menjaga suhu tetap stabil. Selain itu, desain juga memperhatikan kemudahan akses terhadap port USB pada ESP32 untuk keperluan pemrograman atau pengisian daya baterai. Desain 3D dibuat menggunakan perangkat lunak pemodelan 3D seperti *Autodesk Fusion 360*, sehingga hasil desain dapat dicetak menggunakan *printer 3D*.



Gambar 3.4 Schematic Rangkaian Alat

Berdasarkan gambar schematic rangkaian, data hasil deteksi kecepatan kendaraan dari Jetson Nano atau laptop dikirim ke ESP32 melalui jaringan WiFi lokal dengan IP static yang telah diatur. Selanjutnya, ESP32 meneruskan data tersebut ke rangkaian empat modul LED Matrix MAX7219 yang terhubung secara seri untuk menampilkan running text. Pada rangkaian ini, ESP32 terhubung ke LED Matrix melalui beberapa pin: pin VIN sebagai sumber tegangan, pin GND sebagai ground, pin GPIO 23 ke DIN, pin GPIO 5 ke CS, dan pin GPIO 18 ke CLK, sehingga data dapat diteruskan ke setiap modul LED Matrix secara berurutan.

3.7 Data Pengujian

Pengujian data dilakukan setelah sistem selesai dirancang dan berjalan sesuai kebutuhan, dengan tujuan memperoleh hasil deteksi yang akurat dan efisien. Tahapan ini mengacu pada metode yang digunakan oleh (Alif, 2024), diawali dengan pengolahan dataset sebanyak 1.298 gambar kendaraan (mobil dan motor) yang dianotasi, kemudian dibagi menjadi 70% data latih, 15% validasi, dan 15% pengujian. Model YOLOv11 dilatih selama 300 epoch menggunakan optimizer Stochastic Gradient Descent (SGD) dengan learning rate 0,01, momentum 0,937, dan weight decay 0,0005, serta batch size disesuaikan menjadi 8 dan 16 sesuai

keterbatasan sumber daya. Untuk meningkatkan generalisasi model, diterapkan teknik augmentasi seperti perubahan *hue*, *brightness*, *saturation*, dan mosaik. Evaluasi kinerja model dilakukan menggunakan metrik *precision*, *recall*, *F1-score*, *mAP@0.5*, *mAP@0.5:0.95*, serta waktu inferensi untuk memastikan kemampuan deteksi secara real-time.

Evaluasi dilakukan untuk menganalisis pengaruh jumlah epoch, variasi batch size, jenis optimizer, serta perbandingan antar varian YOLOv11 seperti YOLOv11n, YOLOv11s, dan YOLOv11m, dengan cara menguji setiap parameter secara terpisah sambil mempertahankan variabel lain tetap konstan agar diperoleh konfigurasi training yang optimal, stabil, dan akurat. Pengujian kemampuan deteksi jenis kendaraan dilakukan menggunakan video lalu lintas di area masuk double way Universitas Jember untuk memastikan model dapat membedakan kendaraan roda dua (motor) dan roda empat (mobil). Selain itu, dilakukan juga pengujian estimasi kecepatan kendaraan berdasarkan waktu tempuh antara dua garis bantu (ROI) yang sudah ditentukan pada citra video, dengan perhitungan kecepatan menggunakan rumus :

$$V = s/t \quad (3.1)$$

Keterangan:

V = Kecepatan

s = Jarak sebenarnya antara kedua garis deteksi

t = Waktu yang diperlukan suatu objek untuk melewati kedua garis deteksi

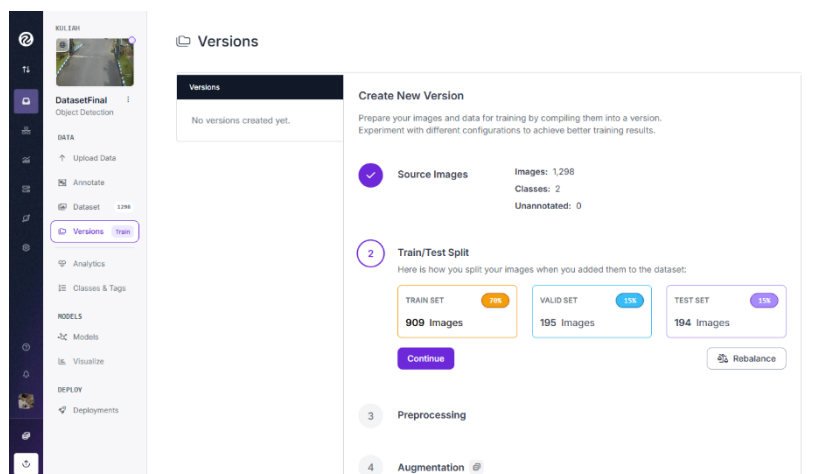
Pengujian dilakukan untuk memastikan hasil deteksi dapat ditampilkan dengan baik pada modul LED MAX7219, meliputi keterbacaan, keandalan komunikasi, serta pengiriman data jenis kendaraan dan kecepatan secara otomatis. Selain itu, pengujian di Jetson Nano mengevaluasi kinerja deteksi kendaraan secara langsung di perangkat *edge* dengan membandingkan estimasi kecepatan dan FPS terhadap laptop atau PC. Terakhir, sistem diuji secara real-time menggunakan video CCTV melalui jaringan IP lokal, di mana model YOLOv11n mendeteksi jenis, jumlah kendaraan, serta mengestimasi kecepatan yang divalidasi dengan *Speed Gun*. Keseluruhan pengujian ini memastikan sistem dapat bekerja langsung dan stabil dalam kondisi *real time*.

BAB 4. PEMBAHASAN

Bab ini membahas hasil pengujian sistem pendeteksi kecepatan kendaraan menggunakan model YOLOv11 berbasis *Convolutional Neural Network* (CNN), yang dikembangkan di Laboratorium CDAST Universitas Jember.

4.1 Pengolahan Dataset dan Parameter *Training*

Pada tahap awal, proses pengolahan data dilakukan dengan membagi dataset menjadi tiga bagian, yaitu 70% untuk data pelatihan (*training set*), 15% untuk data validasi (*validation set*), dan 15% untuk data pengujian (*test set*). Total gambar yang digunakan adalah sebanyak 1.298 citra, yang terdiri dari 909 gambar untuk *training*, 195 gambar untuk *validation*, dan 194 gambar untuk *test*. Dataset ini diperoleh dari rekaman kamera CCTV dengan objek berupa kendaraan yang bergerak di arah masuk *double way* Universitas Jember. Proses anotasi atau pelabelan dilakukan untuk dua kelas, yaitu mobil dan motor, dengan menggunakan platform *Roboflow* untuk membantu dalam *labeling* dan pengelolaan dataset.



Gambar 4.1 *Labeling* Dataset pada *Roboflow*

Training model dilakukan menggunakan YOLOv11 dengan framework *PyTorch*, dan pengujian dilakukan di lingkungan *Kaggle Notebook*. Perintah *training* dijalankan menggunakan skrip `!yolo task=detect mode=train`, dengan

model dasar yolo11n.pt. Dataset dikonfigurasi melalui file data.yaml yang berisi informasi tentang kelas dan direktori data. Parameter *training* yang digunakan meliputi 300 *epoch*, ukuran *batch* 8, dan ukuran gambar 640x640 piksel. *Optimizer* yang digunakan adalah SGD dengan nilai *learning rate* awal (*lr0*) sebesar 0.01, *momentum* sebesar 0.937, serta *weight decay* sebesar 0.0005. Selain itu, augmentasi citra seperti *hsv_h* = 0.015, *hsv_s* = 0.7, *hsv_v* = 0.4, dan *perspective* = 0.001 diterapkan. Namun, parameter *mosaic* dan *close_mosaic* diset ke nol untuk menonaktifkan teknik augmentasi mosaik, karena kamera pengawasan bersifat statis.

Dalam proses evaluasi performa model deteksi objek seperti YOLOv11, digunakan beberapa metrik evaluasi, antara lain *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). *True Positive* (TP) terjadi ketika model berhasil mendeteksi objek yang memang ada dan sesuai kelasnya. *False Positive* (FP) terjadi saat model mendeteksi objek yang sebenarnya tidak ada, *True Negative* (TN) terjadi ketika model tidak mendeteksi objek dan memang tidak ada objek yang seharusnya terdeteksi, sehingga keputusan model dapat dianggap benar, sedangkan *False Negative* (FN) adalah ketika model gagal mendeteksi objek yang sebenarnya ada. Berdasarkan metrik ini, *precision* dihitung dengan rumus:

$$precision = \frac{TP}{TP + FP} \quad (4.1)$$

yang menunjukkan seberapa banyak prediksi yang benar dari seluruh hasil deteksi model. Sedangkan *recall* dihitung dengan rumus:

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

yang mengukur kemampuan model dalam menemukan seluruh objek yang sebenarnya. Kemudian, untuk F1 Score, metrik ini digunakan untuk mengevaluasi keseimbangan antara Precision dan Recall dengan cara menggabungkan keduanya ke dalam satu nilai tunggal yang merepresentasikan trade-off antara kedua metrik tersebut. Rumus F1 Score ditunjukkan pada Persamaan (4.3) berikut:

$$F1\ Score = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (4.3)$$

4.2 Hasil Evaluasi Parameter dan Pengujian YOLOv11

Membahas hasil evaluasi terhadap parameter *training* dan performa model deteksi menggunakan YOLOv11. Ini mencakup pengujian terhadap:

4.2.1 Pengaruh Jumlah *Epoch* Terhadap Nilai Akurasi Model

Epoch adalah parameter yang menentukan seberapa banyak proses pelatihan (*training*) diulang terhadap dataset yang telah dikumpulkan, dalam hal ini dilakukan menggunakan *Kaggle Notebook*. *Epoch* merupakan kondisi yang melalui *training* yang telah di update dalam satu siklus penuh. Penelitian yang dilakukan ini menggunakan beberapa pengujian dengan mengubah nilai *Epoch* dalam proses *training* diantaranya yaitu 100 dan 300. Pengujian ini bertujuan untuk mengetahui pengaruh nilai *Epoch* terhadap mAP yang dihasilkan. Berikut merupakan pengujian dari nilai *Epoch* dapat dilihat pada tabel 4.1 Hasil pengujian nilai *Epoch* dibawah ini:

Tabel 4.1 Hasil pengujian nilai *Epoch*

| NO | IMG | Batch | Epoch | Model | Optimizer | mAP50 | mAP50-95 |
|----|-----|-------|-------|----------|-----------|---------|----------|
| 1 | 640 | 8 | 100 | YOLOV11n | SGD | 0,99077 | 0,78643 |
| 2 | 640 | 8 | 300 | YOLOV11n | SGD | 0,99000 | 0,79482 |

Berdasarkan Tabel 4.1, peningkatan epoch dari 100 ke 300 berpengaruh pada nilai mAP50-95, meskipun mAP50 sedikit menurun. Pada *epoch* ke-100, mAP50 sebesar 0,99077 dan mAP50-95 sebesar 0,78643; sementara pada *epoch* ke-300, mAP50 turun menjadi 0,99000, namun mAP50-95 meningkat menjadi 0,79482. Artinya, penambahan *epoch* tidak selalu meningkatkan akurasi pada *threshold* IoU 50% (mAP50), tetapi dapat memperkuat kemampuan generalisasi model pada berbagai tingkat *overlap* (mAP50-95), sehingga model menjadi akurat terhadap variasi objek.

4.2.2 Evaluasi *Batch Size* terhadap Akurasi Deteksi

Batch size merupakan parameter penting dalam *training* model *deep learning* karena memengaruhi stabilitas, konvergensi, dan akurasi deteksi. Pada subbab ini, dilakukan pengujian untuk mengevaluasi pengaruh variasi *batch size* terhadap

performa deteksi model YOLOv11 terhadap kendaraan. Pengujian dilakukan dengan variasi *batch size*, sementara parameter lain seperti *epoch*, ukuran gambar, dan *optimizer* dibuat tetap. Hasil pengujian disajikan pada Tabel 4.2.

Tabel 4.2 Hasil pengujian nilai *Batch*

| NO | IMG | Epoch | Batch | Model | Optimizer | mAP50 | mAP50-95 |
|----|-----|-------|-------|----------|-----------|---------|----------|
| 1 | 640 | 300 | 8 | YOLOV11n | SGD | 0,99000 | 0,79482 |
| 2 | 640 | 300 | 16 | YOLOV11n | SGD | 0,99113 | 0,79889 |

Hasil pada Tabel 4.2 menunjukkan bahwa variasi *batch size* memengaruhi akurasi deteksi model. Dengan parameter lainnya tetap, YOLOv11n menghasilkan mAP50 sebesar 0,99000 dan mAP50-95 sebesar 0,79482 pada *batch size* 8. Saat *batch size* ditingkatkan menjadi 16, nilai mAP50 naik menjadi 0,99113 dan mAP50-95 menjadi 0,79889. Ini menunjukkan bahwa *batch size* lebih besar memberikan pembaruan bobot yang lebih stabil dan performa deteksi yang sedikit lebih baik. Meskipun selisihnya kecil, pemilihan *batch size* tetap berkontribusi pada efektivitas *training*.

4.2.3 Perbandingan *Optimizer* terhadap Konvergensi Model

Optimasi merupakan bagian penting dalam *training* jaringan syaraf tiruan untuk memperoleh bobot yang menghasilkan *output* terbaik. Dalam penelitian ini digunakan dua jenis *optimizer*, yaitu *Stochastic Gradient Descent* (SGD) dan *Adaptive Moment Estimation* (Adam). SGD bekerja dengan memperbarui bobot secara bertahap setiap kali memproses satu data, sehingga model dapat terus belajar tanpa menunggu seluruh epoch selesai.

Sementara itu, Adam merupakan *optimizer* adaptif yang menggabungkan keunggulan *AdaGrad* dan *RMSProp*. Dengan menggunakan rata-rata *momentum* dan skala gradien terkini, Adam mampu menyesuaikan learning rate secara otomatis dan mempercepat konvergensi, terutama pada data dengan *noise* tinggi. Kedua *optimizer* ini diuji untuk melihat pengaruhnya terhadap akurasi dan performa model, yang hasilnya ditampilkan pada Tabel 4.3.

Tabel 4.3 Hasil pengujian *optimizer*

| NO | IMG | Epoch | Batch | Model | Optimizer | mAP50 | mAP50-95 |
|----|-----|-------|-------|----------|-----------|---------|----------|
| 1 | 640 | 300 | 8 | YOLOV11n | AdamW | 0,99150 | 0,76437 |
| 2 | 640 | 300 | 8 | YOLOV11n | SGD | 0,99000 | 0,79482 |

Pengujian dilakukan menggunakan dua jenis optimizer, yaitu Adam dan SGD (*Stochastic Gradient Descent*), dengan evaluasi berdasarkan nilai mAP. *Optimizer* Adam menghasilkan mAP sebesar 0,99159 melalui pembaruan bobot secara adaptif setiap *epoch*. Sementara itu, penggunaan SGD pada percobaan kedua memberikan hasil lebih baik dengan mAP sebesar 0,99219, karena melakukan pembaruan bobot setiap satu data, sehingga klasifikasi lebih akurat dan responsif. Berdasarkan hasil tersebut, dapat disimpulkan bahwa *optimizer* SGD memberikan performa training yang lebih baik dibandingkan Adam.

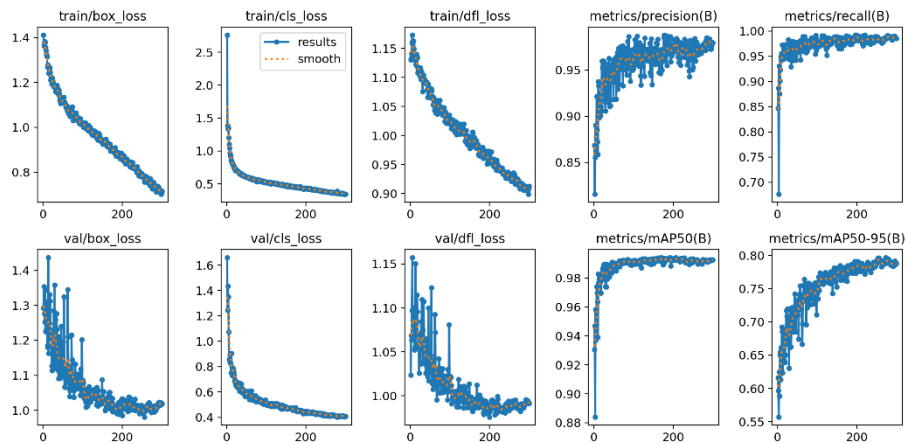
4.2.4 Perbandingan Model YOLOv11 terhadap Akurasi dan Efisiensi

Untuk memperoleh model deteksi yang optimal, tidak hanya diperlukan akurasi yang tinggi, tetapi juga efisiensi dalam waktu inferensi agar dapat digunakan secara real-time. Oleh karena itu, pada subbab ini dilakukan pengujian terhadap beberapa varian model dari YOLOv11, yaitu YOLOv11n (nano), YOLOv11s (small), dan YOLOv11m (medium). Ketiga varian tersebut diuji dengan konfigurasi parameter yang sama, yaitu ukuran gambar 640, jumlah epoch sebanyak 300, batch size 8, dan optimizer SGD. Pengujian ini bertujuan untuk membandingkan performa akurasi deteksi yang diwakili oleh nilai mAP50, serta efisiensi dalam hal kecepatan pemrosesan yang diukur melalui waktu inferensi rata-rata per frame (dalam satuan milidetik). Hasil pengujian ini diharapkan membantu memilih model terbaik untuk sistem deteksi kecepatan kendaraan, yang dirangkum pada Tabel 4.4.

Tabel 4.4 Hasil pengujian model YOLOV11

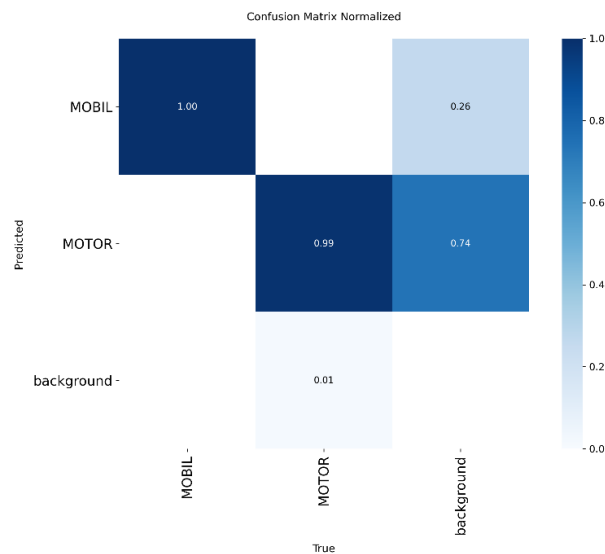
| No | Img | Epoch | Batch | Model | Optimizer | mAP50 | Inference |
|----|-----|-------|-------|----------|-----------|---------|-----------|
| 1 | 640 | 300 | 8 | YOLOV11n | SGD | 0,99000 | 3,2 |
| 2 | 640 | 300 | 8 | YOLOV11s | SGD | 0,99189 | 4,5 |
| 3 | 640 | 300 | 8 | YOLOV11m | SGD | 0,99274 | 11,5 |

a. Hasil Pengujian model YOLOV11n



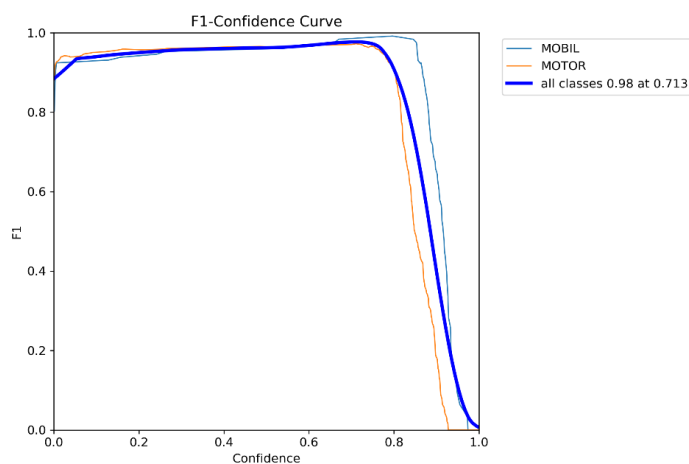
Gambar 4.2 Hasil training menggunakan model YOLOV11n

Training model YOLOv11n menunjukkan tren konvergen yang stabil. Nilai *box_loss*, *cls_loss*, dan *dfl_loss* terus menurun hingga akhir *epoch* ke-300, menandakan bahwa model mampu belajar dengan baik dari data yang digunakan. Evaluasi terhadap metrik validasi juga memperlihatkan peningkatan secara konsisten. Nilai mAP50 mencapai 0.990, yang mengindikasikan akurasi deteksi yang sangat tinggi pada ambang *threshold* 0.5. Meskipun YOLOv11n termasuk dalam varian model ringan, performanya tetap bagus dengan presisi, *recall*, dan stabilitas *loss* yang baik selama proses *training* dan validasi.



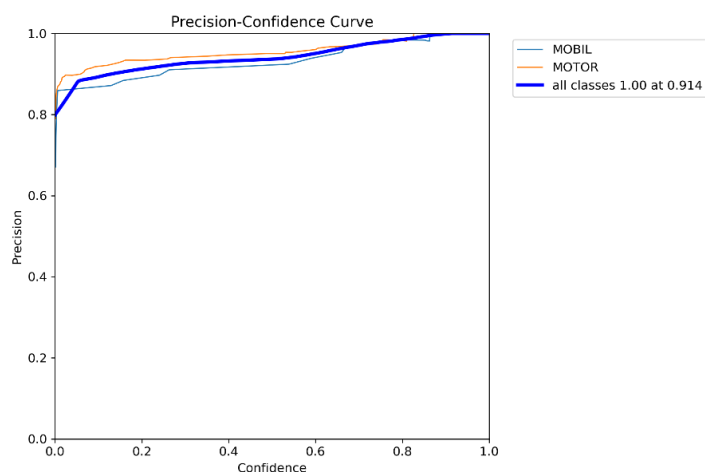
Gambar 4.3 Confusion matrix

Gambar 4.3 *Confusion matrix* terstandarisasi menunjukkan bahwa model mampu mengklasifikasikan objek MOBIL dengan sempurna (akurasi 100%) dan MOTOR dengan tingkat kesalahan yang relatif kecil. Terdapat sedikit *false positive* dari kelas *background* yang terdeteksi sebagai MOBIL sebesar 0.26, serta prediksi MOTOR yang cukup dominan pada kedua kelas utama, yaitu sebesar 0.74.



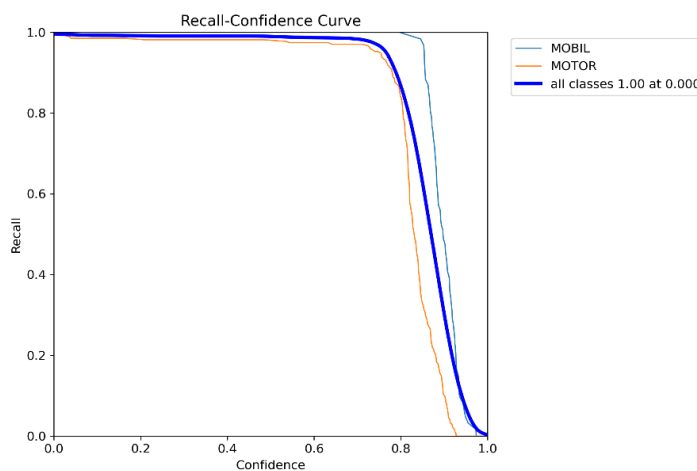
Gambar 4.4 Kurva nilai F1 terhadap nilai *Confidence*

Gambar 4.4 Kurva F1 terhadap *confidence* menunjukkan nilai *F1-score* yang tinggi dan stabil di berbagai tingkat *confidence*. Nilai puncaknya tercapai pada *confidence* 0.713 dengan *F1-score* rata-rata sebesar 0.98 untuk semua kelas, menunjukkan keseimbangan yang sangat baik antara *precision* dan *recall*. Hal ini menandakan bahwa model memiliki kemampuan dalam mendeteksi objek dengan minim kesalahan.



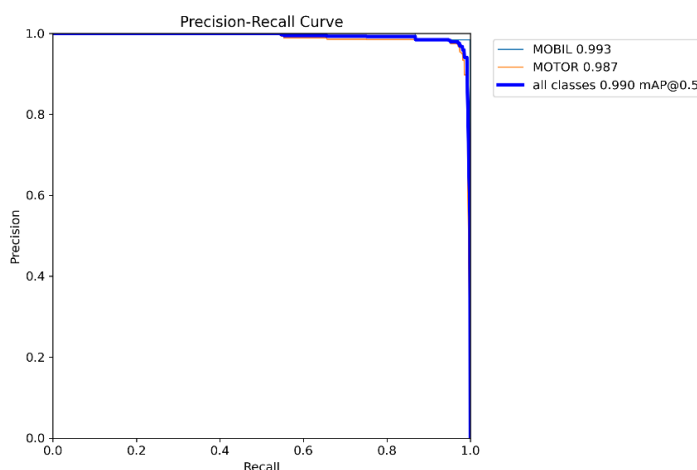
Gambar 4.5 Kurva nilai *Precision* terhadap nilai *Confidence*

Gambar 4.5 Kurva menunjukkan precision meningkat seiring naiknya confidence threshold, dengan precision tertinggi mendekati 1.00 pada confidence 0.914. Hal ini menandakan bahwa saat model sangat yakin, hampir semua prediksinya benar, cocok untuk aplikasi yang memerlukan akurasi tinggi.



Gambar 4.6 Kurva nilai *Recall* terhadap nilai *Confidence*

Gambar 4.6 Kurva pada gambar menunjukkan recall tetap tinggi mendekati 1.00 di confidence rendah, dan mulai menurun secara bertahap saat confidence meningkat, menandakan deteksi yang stabil untuk semua kelas.

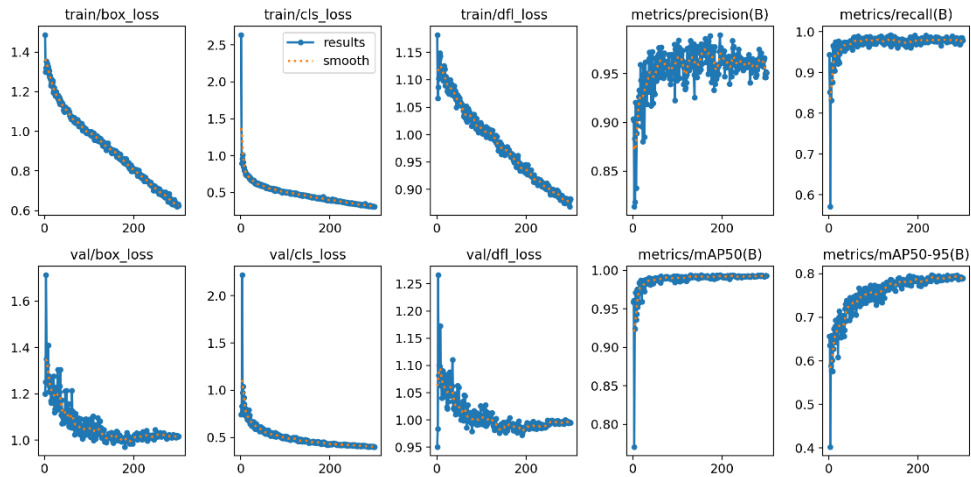


Gambar 4.7 Kurva nilai *Precision* terhadap nilai *Recall*

Gambar 4.7 menunjukkan kurva Precision-Recall yang mengonfirmasi akurasi deteksi tinggi untuk kedua kelas (MOBIL dan MOTOR), dengan mAP50 sebesar 0.990. Precision tercatat 0.993 untuk MOBIL dan 0.987 untuk MOTOR.

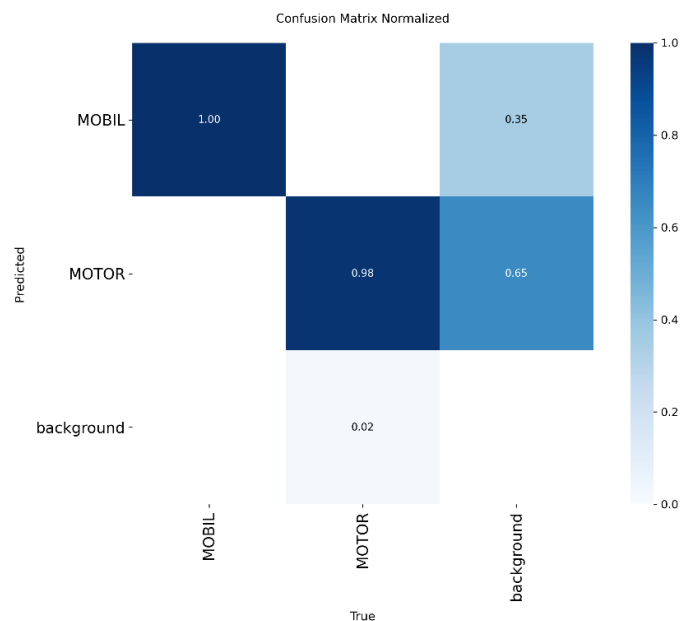
Hasil ini menegaskan bahwa meskipun YOLOv11n merupakan varian paling ringan, performanya tetap sangat baik.

b. Hasil Pengujian model YOLOV11s



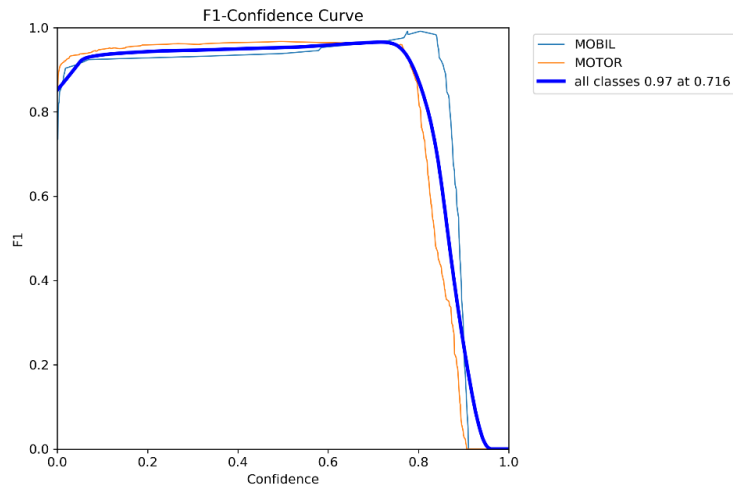
Gambar 4.8 Hasil *training* menggunakan model YOLOV11s

Training YOLOv11s menunjukkan penurunan stabil pada `box_loss`, `cls_loss`, dan `df_l_loss`, dari 1.4 ke sekitar 0.6, menandakan pembelajaran yang baik. Evaluasi menghasilkan mAP50 sebesar 0.992, dengan presisi dan recall yang terus meningkat hingga 300 epoch. Model ini cepat konvergen dan tetap berkinerja tinggi meski termasuk model ringan.



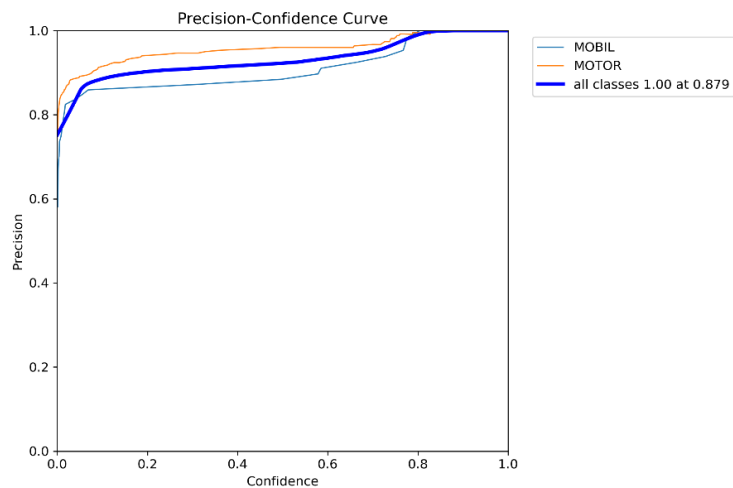
Gambar 4.9 *Confusion matrix*

Gambar 4.9 menunjukkan model mendeteksi MOBIL dengan akurasi 100%, meski ada *overlap background* yang terdeteksi sebagai MOBIL (0.35) dan sedikit *false positive* pada MOTOR (0.65). Prediksi kelas MOTOR tetap tinggi (0.98), menandakan YOLOv11s tetap efisien membedakan objek meskipun ada sedikit *noise*.



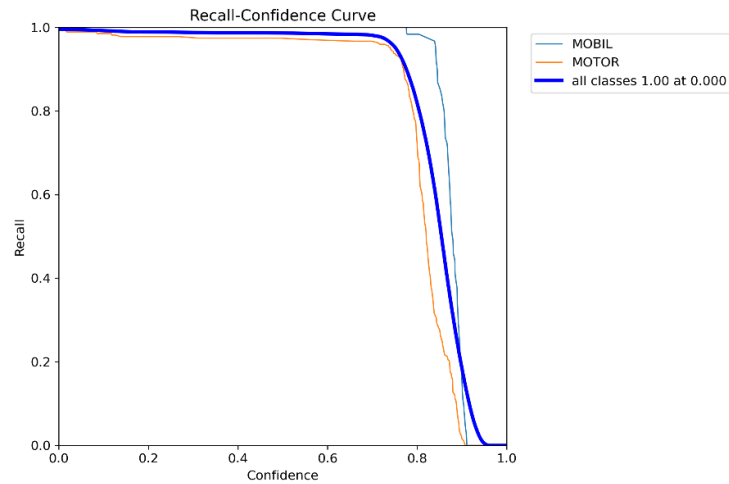
Gambar 4.10 Kurva nilai F1 terhadap nilai *Confidence*

Gambar 4.10 Kurva F1 menunjukkan performa yang sangat kuat dari model, dengan nilai F1-score tertinggi sebesar 0.97 pada *confidence* 0.716 untuk seluruh kelas. Kurva untuk kelas MOBIL dan MOTOR menunjukkan kestabilan pada F1 di atas 0.9 di berbagai tingkat *confidence*, yang berarti model mencapai keseimbangan yang baik antara presisi dan *recall*.



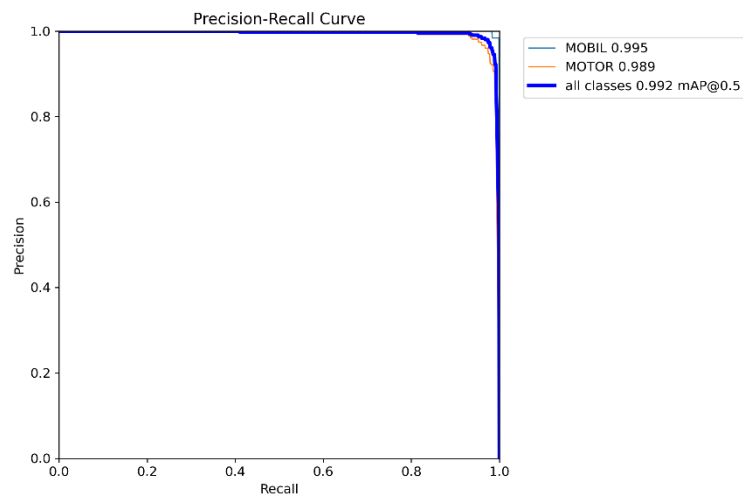
Gambar 4.11 Kurva nilai *Precision* terhadap nilai *Confidence*

Gambar 4.11 Kurva ini menunjukkan precision mencapai nilai sempurna 1.00 pada *confidence* 0.879. Kelas MOTOR memiliki precision lebih tinggi daripada kelas MOBIL di sebagian besar rentang *confidence*. Ini mengindikasikan bahwa saat model lebih yakin terhadap prediksinya, hampir seluruh prediksi tersebut benar. Hal ini sangat bermanfaat untuk aplikasi yang membutuhkan presisi tinggi.



Gambar 4.12 Kurva nilai *Recall* terhadap nilai *Confidence*

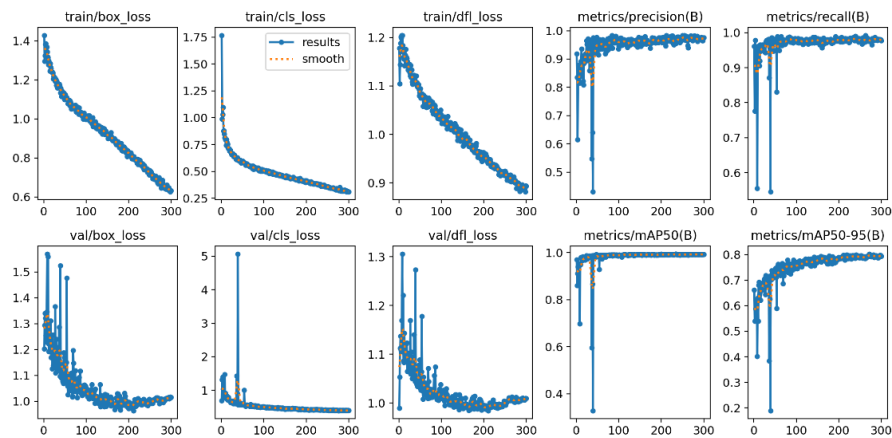
Gambar 4.12 *Recall* tertinggi menyentuh nilai 1.00 pada *confidence* 0, yang berarti model mampu menangkap seluruh objek pada *threshold* rendah. Penurunan *recall* terjadi seiring meningkatnya *confidence threshold*, terutama pada MOTOR, tetapi penurunannya tetap dalam batas wajar. Ini menunjukkan stabilitas deteksi yang cukup baik bahkan pada model dengan arsitektur kecil.



Gambar 4.13 Kurva nilai *Precision* terhadap nilai *Recall*

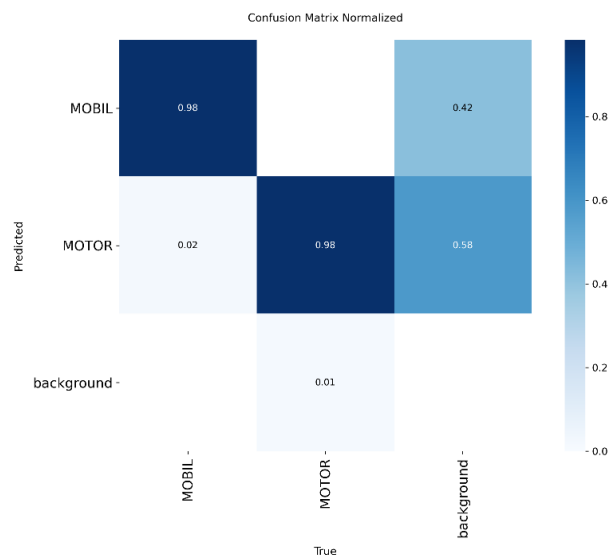
Gambar 4.13 Kurva *Precision-Recall* untuk kedua kelas memperlihatkan nilai *precision* dan *recall* mendekati sempurna. Nilai mAP50 mencapai 0.992, dengan *precision* sebesar 0.995 untuk MOBIL dan 0.989 untuk MOTOR. Ini mengonfirmasi bahwa YOLOv11s memiliki akurasi deteksi yang sangat baik, serta menjaga performa yang kompetitif meskipun termasuk dalam varian ringan.

c. Hasil Pengujian model YOLOV11m



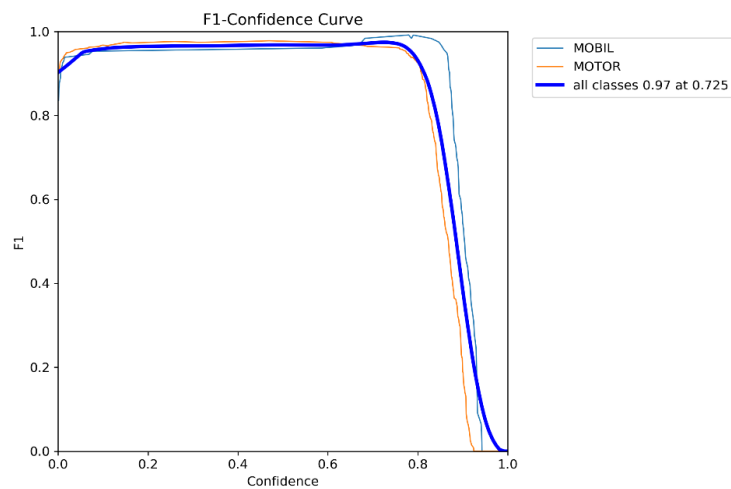
Gambar 4.14 Hasil training menggunakan model YOLOV11m

Gambar 4.14 Hasil training YOLOv11m menunjukkan konvergensi baik hingga epoch ke-300, dengan penurunan stabil pada box_loss, cls_loss, dan df_loss, serta mAP50 mendekati 1.00 dan peningkatan precision dan recall yang signifikan.



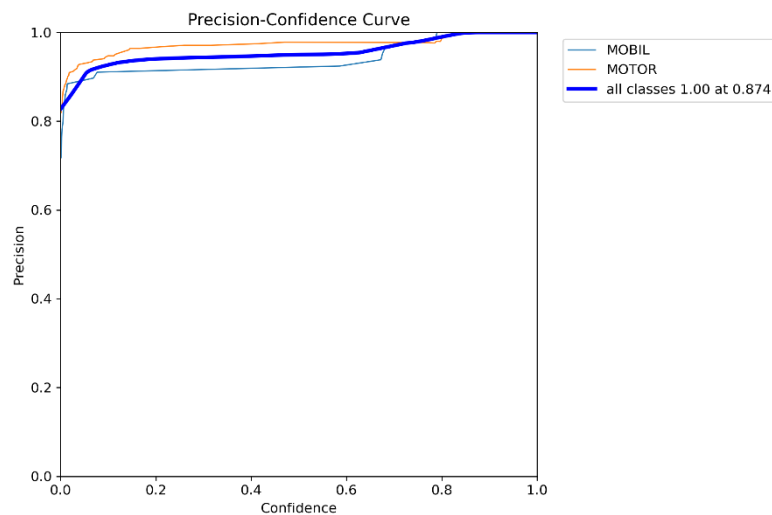
Gambar 4.15 *Confusion matrix*

Gambar 4.15 *Confusion matrix* memperlihatkan bahwa model memiliki akurasi tinggi dalam membedakan objek MOBIL dan MOTOR, dengan akurasi masing-masing sebesar 0.98 dan 0.98. Terdapat *false positive* dari kelas *background* yang diklasifikasikan sebagai MOBIL sebesar 0.42, serta sebagai MOTOR sebesar 0.58. Namun, nilai ini masih dalam batas toleransi untuk model menengah dan tidak mengganggu klasifikasi yang tepat oleh YOLOv11m.



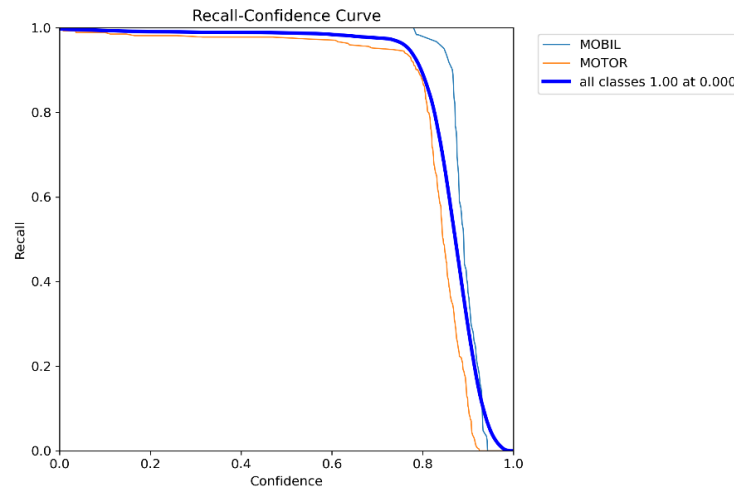
Gambar 4.16 Kurva nilai F1 terhadap nilai *Confidence*

Gambar 4.16 Kurva F1 terhadap *confidence* menunjukkan performa yang sangat stabil, dengan nilai puncak F1-score sebesar 0.97 pada confidence 0.725. Hal ini mencerminkan keseimbangan optimal antara precision dan recall, bahkan pada berbagai nilai *confidence*.



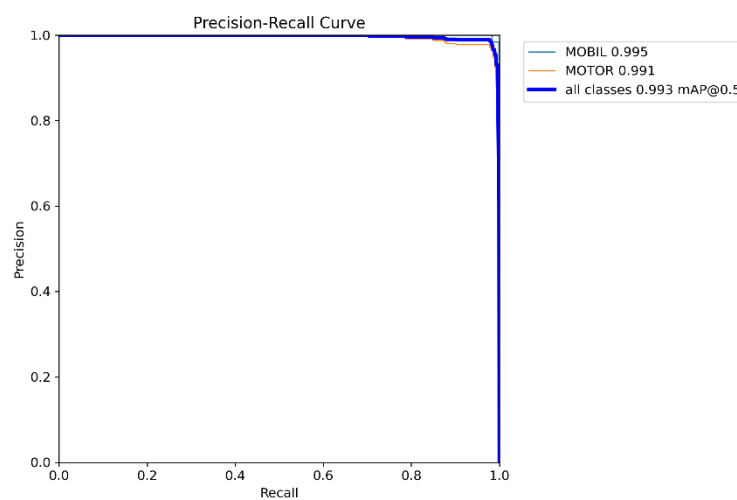
Gambar 4.17 Kurva nilai *Precision* terhadap nilai *Confidence*

Gambar 4.17 Kurva *precision* terhadap *confidence* menunjukkan bahwa *precision* meningkat seiring dengan *confidence*. *Precision* maksimum mencapai 1.00 pada *confidence* 0.874, menandakan bahwa ketika model cukup yakin, maka prediksi yang dihasilkan hampir sepenuhnya benar. Ini sangat cocok untuk aplikasi yang membutuhkan keputusan akurat saat *confidence* tinggi.



Gambar 4.18 Kurva nilai *Recall* terhadap nilai *Confidence*

Gambar 4.18 Kurva recall menunjukkan bahwa recall tetap tinggi bahkan pada *confidence* rendah, dan mencapai nilai maksimum 1.00 pada *confidence* 0. Meskipun terjadi penurunan *recall* di *confidence* tinggi (wajar), penurunannya tidak tajam, menandakan bahwa model mampu mempertahankan deteksi menyeluruh secara konsisten.



Gambar 4.19 Kurva nilai *Precision* terhadap nilai *Recall*

Gambar 4.19 Kurva *precision-recall* mengonfirmasi performa deteksi yang sangat baik untuk kedua kelas (MOBIL dan MOTOR), dengan nilai *precision* dan *recall* yang mendekati sempurna. *Precision* sebesar 0.995 untuk MOBIL dan 0.991 untuk MOTOR, menghasilkan $mAP@0.5$ sebesar 0.993. Kinerja ini memperkuat bahwa YOLOv11m memiliki keseimbangan antara *precision* dan *recall* yang ideal, menjadikannya cocok untuk kebutuhan deteksi akurat dan sensitif secara bersamaan.

Dari hasil pengujian yang telah dilakukan terhadap ketiga varian model YOLOv11, terlihat bahwa setiap model menunjukkan performa deteksi yang sangat baik dengan nilai $mAP50$ di atas 0.990. Meskipun YOLOv11m mencatatkan nilai $mAP50$ tertinggi sebesar 0.993 dan diikuti oleh YOLOv11s dengan 0.991, selisih akurasi tersebut relatif kecil dibandingkan dengan hasil dari YOLOv11n yang mencapai 0.990. Performa ini menunjukkan bahwa meskipun YOLOv11n merupakan varian paling ringan secara arsitektur, akurasi deteksinya tetap kompetitif. Selain itu, waktu inferensi dari YOLOv11n yang jauh lebih cepat, yaitu hanya 3.2 ms per frame, menjadi keunggulan signifikan dalam konteks penerapan di perangkat dengan sumber daya terbatas seperti Jetson Nano. Faktor-faktor ini menunjukkan bahwa efisiensi waktu proses dan kemudahan komputasi dapat dicapai tanpa mengorbankan akurasi secara signifikan, menjadikan varian ini sangat sesuai untuk kebutuhan deteksi *real-time* dengan keterbatasan perangkat keras.

4.3 Pengujian Pendeteksian Jenis dan Jumlah Kendaraan

Pengujian ini bertujuan untuk mengevaluasi kemampuan sistem dalam mendeteksi dan mengklasifikasikan jenis serta jumlah kendaraan yang melintas di depan kamera, khususnya antara kendaraan roda dua (motor) dan roda empat (mobil). Data pengujian diperoleh dari rekaman video CCTV yang merekam arus lalu lintas secara *real-time* di lingkungan Universitas Jember. Sistem akan membandingkan hasil deteksi otomatis dari model YOLOv11 dengan hasil hitung manual yang dijadikan sebagai acuan (*ground truth*). Perbandingan ini bertujuan untuk mengukur akurasi deteksi jenis kendaraan dan menghitung persentase error

antara deteksi sistem dan perhitungan manual untuk masing-masing jenis kendaraan. Hasil pengujian disajikan pada Tabel 4.5 berikut:

Tabel 4.5 Hasil pengujian pendeteksian kendaraan

| Pengujian Video ke- | Waktu | Hitung Manual | | Hasil Deteksi | | Error Persen Mobil (%) | Error Persen Motor (%) |
|---------------------|-------|---------------|-------|---------------|-------|------------------------|------------------------|
| | | Motor | Mobil | Motor | Mobil | | |
| 1 | Pagi | 9 | 2 | 8 | 2 | 0 | 11,11 |
| 2 | Pagi | 24 | 8 | 22 | 8 | 0 | 8,33 |
| 3 | Siang | 62 | 2 | 60 | 2 | 0 | 3,23 |
| 4 | Sore | 88 | 6 | 86 | 6 | 0 | 2,27 |
| Rata-rata | | | | | | 0 | 6,23 |
| Standar Deviasi | | | | | | 0 | 4,20 |
| Koefisien Korelasi | | | | | | 1,00 | 0,99 |

Berdasarkan hasil pengujian pada Tabel 4.5, sistem menunjukkan performa sangat baik dalam mendeteksi jumlah kendaraan jenis mobil, dengan error 0% pada seluruh pengujian, yang mengindikasikan mobil lebih mudah dikenali oleh model. Untuk motor, rata-rata error sebesar 6,23% dengan standar deviasi 4,20% menunjukkan ada sedikit variasi hasil deteksi, yang salah satunya dipengaruhi oleh waktu pengambilan data; pada pagi hari error cenderung lebih tinggi dibandingkan siang dan sore hari akibat kondisi pencahayaan dan bayangan yang lebih tajam, sedangkan pada siang dan sore hari error menurun karena pencahayaan lebih stabil. Selain itu, koefisien korelasi sebesar 1,00 untuk mobil dan 0,99 untuk motor menegaskan bahwa hasil deteksi sangat searah dengan data manual. Secara keseluruhan, model YOLOv11 tetap mampu mendeteksi jenis dan jumlah kendaraan dengan akurasi dan konsistensi yang baik.

4.4 Pengujian Estimasi Kecepatan Kendaraan

Pengujian ini bertujuan untuk mengevaluasi tingkat akurasi sistem dalam mendeteksi kecepatan kendaraan berdasarkan citra/video yang dianalisis oleh model deteksi dan perhitungan kecepatan. Proses pengujian dilakukan dengan membandingkan hasil deteksi kecepatan oleh sistem dengan pembacaan kecepatan aktual dari perangkat *speed gun* sebagai acuan.

Gambar 4.20 *Speed Gun*

Pengujian dilakukan sebanyak sepuluh kali, masing-masing melibatkan kendaraan jenis mobil dan motor, dengan kecepatan yang divariasikan.

Tabel 4.6 Hasil pengujian kecepatan kendaraan

| Pengujian ke | Jenis Kendaraan | Kecepatan Pada Speed Gun (Km/Jam) | Hasil Deteksi Kecepatan (Km/Jam) | Error Persen (%) |
|------------------------------------|-----------------|-----------------------------------|----------------------------------|------------------|
| 1 | Mobil | 27 | 27.59 | 2.19 |
| 2 | Mobil | 25 | 23.89 | 4.44 |
| 3 | Motor | 25 | 24.92 | 0.32 |
| 4 | Motor | 24 | 24.42 | 1.75 |
| 5 | Mobil | 25 | 25.32 | 1.28 |
| 6 | Mobil | 26 | 25.51 | 1.88 |
| 7 | Motor | 24 | 24.42 | 1.75 |
| 8 | Motor | 25 | 24.99 | 0.04 |
| 9 | Motor | 25 | 24.53 | 1.88 |
| 10 | Motor | 26 | 25.76 | 0.92 |
| Rata-rata Error persen motor | | | | 1.11 |
| Rata-rata Error persen mobil | | | | 2.45 |
| Standar Deviasi Error persen motor | | | | 0.80 |
| Standar Deviasi Error persen mobil | | | | 1.38 |
| Koefisien Korelasi motor | | | | 0,91 |
| Koefisien Korelasi mobil | | | | 0,90 |



Hasil pengujian pada Tabel 4.6 menunjukkan bahwa kecepatan kendaraan yang terdeteksi oleh sistem cukup mendekati nilai sebenarnya. Untuk mobil, error persentase tercatat bervariasi antara 1,28% hingga 4,44%, sedangkan pada motor umumnya lebih rendah, selalu di bawah 2% dengan nilai terkecil mencapai 0,04%. Secara keseluruhan, rata-rata error persen adalah 1,11% untuk motor dan 2,45%

untuk mobil, dengan standar deviasi masing-masing sebesar 0,80% dan 1,38%, yang menunjukkan hasil deteksi cukup konsisten. Selain itu, nilai koefisien korelasi sebesar 0,91 untuk motor dan 0,90 untuk mobil mengindikasikan hubungan yang kuat antara hasil deteksi sistem dan data speed gun sebagai acuan. Temuan ini memperkuat bahwa sistem mampu memperkirakan kecepatan kendaraan dengan akurasi dan stabilitas yang baik, khususnya pada kendaraan jenis motor.

4.5 Pengujian Tampilan *Running Text*

Pengujian tampilan running text dilakukan untuk memastikan hasil prediksi kecepatan kendaraan dapat dikirim dan ditampilkan dengan baik pada modul LED Matrix. Hal ini penting agar informasi kecepatan tidak hanya tersimpan dalam sistem, tetapi juga dapat dilihat langsung oleh pengguna sebagai umpan balik real-time. Pengujian dilakukan sebanyak lima kali dengan variasi kecepatan kendaraan, dan hasil deteksi berhasil tampil sesuai data yang terdeteksi.

Tabel 4.7 Hasil pengujian *running text*

| Pengujian Ke- | Prediksi kecepatan | Tingkat Keberhasilan | Gambar |
|---------------|--------------------|----------------------|---|
| 1 | 27,19 km/jam | Berhasil |  |
| 2 | 29,36 km/jam | Berhasil |  |
| 3 | 35,83 km/jam | Berhasil |  |
| 4 | 38,36 km/jam | Berhasil |  |
| 5 | 44,4 km/jam | Berhasil |  |

Tabel 4.7 menunjukkan seluruh hasil prediksi kecepatan berhasil ditampilkan pada LED Matrix, dengan variasi kecepatan 27,19–44,4 km/jam dan status selalu “Berhasil” tanpa gangguan. Sistem memanfaatkan jaringan WiFi biasa, meskipun

untuk realtime tetap perlu terhubung ke jaringan WiFi UNEJ agar dapat mengakses IP CCTV. Secara keseluruhan, sistem mampu menampilkan informasi kecepatan secara real-time dengan konsistensi baik, mendukung integrasi deteksi dan antarmuka untuk pemantauan kendaraan di lapangan.

4.6 Pengujian Program pada Jetson Nano

Pengujian program pada Jetson Nano dilakukan untuk mengevaluasi sejauh mana perangkat ini mampu menjalankan algoritma deteksi kendaraan berbasis YOLO secara optimal, dengan fokus pada selisih error dibanding hasil deteksi di laptop atau PC. Pengujian juga menilai performa Jetson Nano dalam hal kecepatan frame per second (FPS) selama proses inferensi. Mengingat keterbatasan sumber daya, model yang digunakan adalah YOLOv11n agar dapat dijalankan lebih ringan.

Tabel 4.8 Hasil Pengujian Pada Jetson Nano

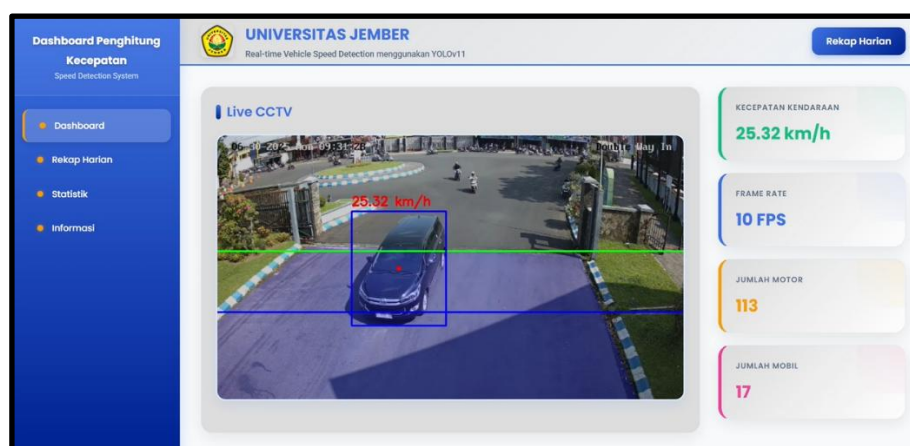
| No | Jenis kendaraan | Hasil Prediksi Yolo kecepatan (Km/jam) | Hasil Kecepatan Jetson Nano (Km/Jam) | Error Persen (%) |
|------------------------------------|-----------------|--|--------------------------------------|------------------|
| 1 | Mobil | 22,99 | 17,8 | 22,57 |
| 2 | Motor | 25,82 | 18,02 | 30,23 |
| 3 | Motor | 25,82 | 17,89 | 30,70 |
| 4 | Motor | 27,15 | 18,45 | 32,04 |
| 5 | Motor | 25,1 | 18,02 | 28,23 |
| 6 | Motor | 26,00 | 18,10 | 30,04 |
| 7 | Motor | 26,10 | 18,20 | 30,44 |
| 8 | Mobil | 23,00 | 17,9 | 22,90 |
| 9 | Mobil | 23,20 | 17,8 | 23,40 |
| 10 | Mobil | 22,80 | 17,7 | 22,40 |
| Rata-rata Error Persen Motor | | | | 30,28 |
| Rata-rata Error Persen Mobil | | | | 22,82 |
| Standar Deviasi Error persen motor | | | | 1,14 |
| Standar Deviasi Error persen mobil | | | | 0,41 |
| Koefisien Korelasi motor | | | | 0,83 |
| Koefisien Korelasi mobil | | | | 0,50 |

Berdasarkan hasil pengujian yang ditampilkan pada Tabel 4.8 menunjukkan bahwa prediksi kecepatan kendaraan oleh Jetson Nano masih memiliki tingkat error yang cukup signifikan, yaitu rata-rata sebesar 22,57% untuk kendaraan mobil dan 30,30% untuk kendaraan motor. Meskipun sistem masih mampu melakukan estimasi kecepatan, tingkat akurasi tersebut menunjukkan bahwa Jetson Nano

belum optimal dalam melakukan inferensi kecepatan secara presisi. Selain itu, dalam aspek visualisasi video secara *real-time*, performa Jetson Nano juga mengalami keterbatasan, seperti FPS yang rendah dan tampilan video yang patah-patah. Masalah ini terutama disebabkan oleh keterbatasan sumber daya komputasi Jetson Nano saat menjalankan model deteksi objek secara langsung menggunakan YOLOv11n.

4.7 Pengujian Sistem secara *Real Time*

Pengujian sistem *real-time* dilakukan menggunakan kamera CCTV yang terhubung ke jaringan lokal melalui alamat IP sebagai sumber input video, kemudian diproses oleh model YOLOv11n untuk mendeteksi jenis kendaraan, menghitung jumlah, dan memperkirakan kecepatan. Pengujian ini mengevaluasi performa sistem dalam kondisi nyata, termasuk variasi pencahayaan, sudut kamera, dan pergerakan kendaraan, dengan hasil yang divalidasi menggunakan alat Speed Gun. Sistem juga dilengkapi tampilan antarmuka berbasis web yang menampilkan video *real-time* lengkap dengan bounding box, label kecepatan, serta rekap data jumlah kendaraan dan kecepatan terakhir, sehingga memudahkan pengguna memantau dan mengevaluasi performa sistem secara langsung.



Gambar 4.21 Tampilan *Website* pada *localhost*

Melalui tampilan antarmuka berbasis web tersebut, pengguna dapat memonitor langsung hasil pendeteksian kendaraan dan perhitungan kecepatan secara *real-time*. Untuk memastikan keandalan sistem, dilakukan pengujian lebih lanjut dengan membandingkan hasil prediksi kecepatan kendaraan yang dihasilkan

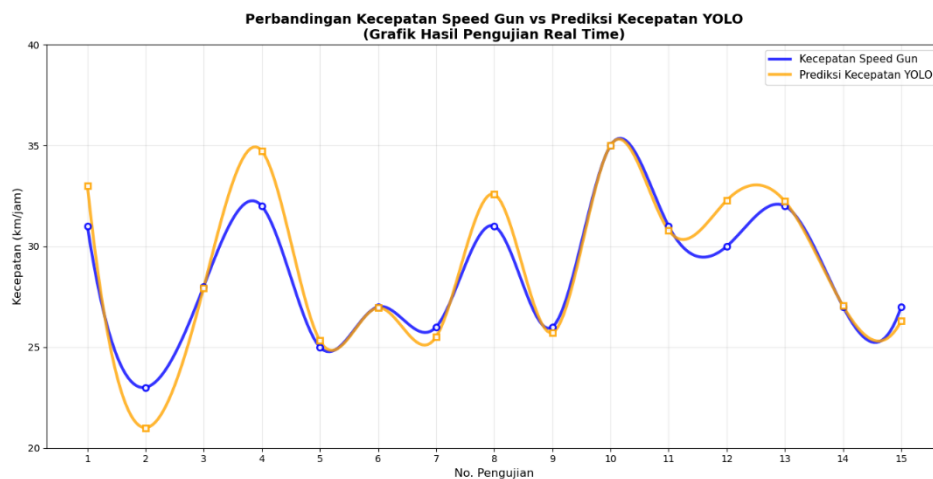
oleh sistem dengan data kecepatan aktual yang diperoleh menggunakan alat Speed Gun. Selain itu, apabila pada saat realtime terjadi kendala seperti fps yang turun, gambar frame yang menumpuk, atau keterlambatan koneksi sehingga sistem tersambung kembali, program tetap dapat melanjutkan proses pendeteksian objek dan jenis kendaraan. Namun, kondisi tersebut menyebabkan data tersebut tidak ikut tercatat dalam counting maupun perhitungan kecepatan, sehingga tidak mempengaruhi akurasi keseluruhan perhitungan yang telah berjalan normal sebelumnya. Hasil pengujian tersebut dirangkum pada Tabel 4.9 berikut.

Tabel 4.9 Hasil Pengujian *Real Time*

| No | Jenis kendaraan | Hasil Kecepatan Speed Gun (Km/Jam) | Hasil Prediksi kecepatan pada YOLO (Km/jam) | Error Persen (%) |
|------------------------------------|-----------------|------------------------------------|---|------------------|
| 1 | Motor | 31 | 33,01 | 6,48 |
| 2 | Mobil | 23 | 21 | 8,70 |
| 3 | Motor | 28 | 27,93 | 0,25 |
| 4 | Motor | 32 | 34,74 | 8,56 |
| 5 | Mobil | 25 | 25,32 | 1,28 |
| 6 | Motor | 27 | 26,95 | 0,19 |
| 7 | Mobil | 26 | 25,51 | 1,88 |
| 8 | Motor | 31 | 32,58 | 5,10 |
| 9 | Mobil | 26 | 25,72 | 1,08 |
| 10 | Motor | 35 | 35,01 | 0,03 |
| 11 | Mobil | 31 | 30,79 | 0,67 |
| 12 | Motor | 30 | 32,28 | 7,60 |
| 13 | Motor | 32 | 32,23 | 0,72 |
| 14 | Motor | 27 | 27,05 | 0,19 |
| 15 | Motor | 27 | 26,29 | 2,63 |
| Rata-rata Error Persen Motor | | | | 3,18 |
| Rata-rata Error Persen Mobil | | | | 2,72 |
| Standar Deviasi Error persen motor | | | | 3,43 |
| Standar Deviasi Error persen mobil | | | | 3,37 |
| Koefisien Korelasi motor | | | | 0,94 |
| Koefisien Korelasi mobil | | | | 0,98 |

Berdasarkan hasil pengujian sistem secara real-time yang ditampilkan pada Tabel 4.9, diperoleh data kecepatan dari 15 kendaraan yang terdiri atas motor dan mobil. Kecepatan aktual diukur menggunakan alat speed gun, kemudian dibandingkan dengan hasil prediksi kecepatan dari sistem berbasis YOLOv11. Untuk kendaraan jenis motor, error persentase bervariasi antara 0,03% hingga

8,56%, dengan rata-rata error sebesar 3,18% dan standar deviasi sebesar 3,43%, menunjukkan variasi hasil prediksi masih tergolong wajar. Sementara itu, untuk mobil, error tertinggi tercatat sebesar 8,70% dan terendah 0,67%, dengan rata-rata error sebesar 2,72% dan standar deviasi 3,37%. Selain itu, nilai koefisien korelasi sebesar 0,94 untuk motor dan 0,98 untuk mobil menunjukkan hubungan yang sangat kuat antara hasil prediksi sistem dan data speed gun, sehingga dapat disimpulkan bahwa sistem mampu memprediksi kecepatan kendaraan secara real-time dengan akurasi dan konsistensi yang baik.



Gambar 4.22 Grafik Hasil Pengujian Sistem secara *Real Time*

Grafik perbandingan hasil kecepatan Speed Gun dan prediksi YOLOv11 menunjukkan pola prediksi yang mendekati data aktual, meskipun ada sedikit selisih di beberapa titik uji. Garis prediksi YOLOv11 cenderung mengikuti tren naik turun hasil Speed Gun, menandakan respons model yang konsisten terhadap perubahan kecepatan kendaraan. Selisih antar garis umumnya kecil dan tetap dalam kisaran wajar, sehingga memperkuat bahwa sistem deteksi kecepatan berbasis YOLOv11 cukup stabil dan layak digunakan secara real-time untuk memantau kecepatan kendaraan.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan dua rumusan masalah yang ditetapkan, diperoleh kesimpulan sebagai berikut:

1. Metode YOLOv11 berhasil mendeteksi dua jenis kendaraan, yaitu mobil dan motor, dengan akurasi yang sangat baik. Hasil pengujian menunjukkan nilai mAP50 di atas 0,99 untuk semua varian model (YOLOv11n, YOLOv11s, dan YOLOv11m). Deteksi jumlah kendaraan juga menunjukkan rata-rata error 0% untuk mobil dan 6,23% untuk motor, dengan standar deviasi error untuk motor sebesar 4,20%, yang menandakan deteksi mobil lebih konsisten karena ukuran objek lebih besar.
2. Sistem yang dibangun menggunakan YOLOv11n mampu memperkirakan kecepatan kendaraan secara real-time dengan akurasi cukup tinggi. Pada pengujian kecepatan, rata-rata error deteksi kecepatan untuk motor adalah 1,11% dan untuk mobil sebesar 2,45%, sedangkan pengujian real-time mencatat rata-rata error 3,18% untuk motor dan 2,72% untuk mobil. Selain itu, seluruh hasil deteksi kecepatan berhasil ditampilkan secara langsung di modul LED Matrix dengan tingkat keberhasilan 100%, sehingga sistem terbukti andal, stabil, dan layak diterapkan untuk pemantauan kecepatan kendaraan di lapangan.

5.2 Saran

Saran untuk pengembangan selanjutnya, terdapat beberapa usulan yaitu:

1. Untuk penelitian selanjutnya dapat menambahkan beberapa kelas lainnya seperti truk, bus dan kendaraan lain.
2. Dapat menggunakan NVIDIA Jetson Orin atau device lain di atasnya yang mendukung untuk menjalankan YOLOv11 keatas.
3. Penelitian selanjutnya dapat mengembangkan sistem deteksi objek pada dua jalur lintasan kendaraan sekaligus dengan memanfaatkan algoritma YOLO versi terbaru untuk meningkatkan akurasi dan efisiensi deteksi.

DAFTAR PUSTAKA

- Alif, M. A. R. (2024). *YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems*. <http://arxiv.org/abs/2410.22898>
- Bamane, K. D., Rajgure, N., Wadne, V., Khaparde, S., Patil, P., Tikait, V., Patankar, A. J., & Gaikwad, A. S. (2024). *Intelligent Systems And Applications In Engineering Enhancement in Real Time Deep Learning Object Detection and Direction Prediction for Visually Impaired using YOLO and OpenCV*. *12*(3), 887–892.
- Erik, M., Rohman, Z., & Editya, A. S. (2025). *Investigasi Forensik Kecelakaan Kendaraan Bermotor Menggunakan Metode Deep Learning*. *9*(4), 6692–6698.
- Faizin, M. (2023). *Implementasi Algoritma YOLO Pada Modul Kamera Untuk Deteksi Jenis Dan Kecepatan Kendaraan*.
- Gibran, H., Purnama, B., & Kosala, G. (2024). *Pengoptimasian Pengukuran Kepadatan Jalan Raya Dengan Cctv Menggunakan Metode Yolov8*. *Technomedia Journal*, *9*(1), 31–45. <https://doi.org/10.33050/tmj.v9i1.2216>
- Hayati, N. J., Singasatia, D., & Muttaqin, M. R. (2023). *Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan*. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, *12*(2), 91–99. <https://doi.org/10.34010/komputa.v12i2.10654>
- Jauhari, N. M. I. (2024). *Sistem Deteksi Kendaraan Menggunakan Metode YOLO V8 Melalui Rekaman CCTV*.
- Juliansyah, R., Aqil, M., & Ar, M. (2024). *Development of a Real-Time Traffic Density Detection Website Using YOLOv8-Based Digital Image Processing with OpenCV*. *6*(4), 2649–2678. <https://doi.org/10.51519/journalisi.v6i4.912>
- Kamble, V. B., Mundhe, O. N., Walunjar, C. M., & Kale, G. A. (2025). *AI-Driven Smart Traffic Management System : An Adaptive Approach Using YOLO and OpenCV*. *2*(2), 66–72. <https://doi.org/10.71141/30485037/V2I2P106>
- Kusuma, A. S., Pradana, A. I., & Pamekas, B. W. (2024). *Pengembangan Sistem Perhitungan Jumlah Kendaraan Berdasarkan Jenis Kendaraan Menggunakan Algoritma YOLO Secara Realtime*. *7*, 166–179.
- Muhlashin, M. N. I., & Stefanie, A. (2023). *Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8*. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(2), 1363–1368. <https://doi.org/10.36040/jati.v7i2.6927>
- Naveenkumar, N., Mahapriya, S., Dhevathai, P., Aswathi, V., & Dhivyadharshini, R. (2025). *Enhancing Deep Learning to Improve Road Safety : An Accident Detection Enhancing Deep Learning to Improve Road Safety : An Accident Detection System*. *April*. <https://doi.org/10.47392/IRJASH.2025.019>
- Pratama, E. W., & Kiswantonono, A. (2023). *Electrical Analysis Using ESP-32 Module In Realtime*. *JEECS (Journal of Electrical Engineering and Computer Sciences)*, *7*(2), 1273–1284. <https://doi.org/10.54732/jeeecs.v7i2.21>
- Sholahuddin, M. R., Harika, M., Awaludin, I., Dewi, Y. C., Dhia Fauzan, F., Sudimulya, B. P., & Widarta, V. P. (2023). *Optimizing YOLOv8 for Real-*

- Time CCTV Surveillance: A Trade-off Between Speed and Accuracy. *Jurnal Online Informatika*, 8(2), 261–270. <https://doi.org/10.15575/join.v8i2.1196>
- Susanto, M. D. (2020). *Rancang Bangun Modul Kamera Pengukur Kecepatan Kendaraan Menggunakan Metode Background Subtraction*.
- Talaat, F. M., El-balka, R. M., Sweidan, S., Gamel, S. A., & Al-, A. M. (2025). *Smart Traffic Management System Using YOLOv11 for Real-Time Vehicle Detection and Dynamic Flow Optimization in Smart Cities*. 1–17.
- Tejashree, K., Ashish, P., Gorai, R., Sah, R., & Kumar, S. (2025). *Speed Detection Of Moving Vehicle Using YOLO. 01*, 1.
- Valladares, S., Toscano, M., Tufiño, R., Morillo, P., & Vallejo-Huanga, D. (2021). *Performance Evaluation of the Nvidia Jetson Nano Through a Real-Time Machine Learning Application*. *January*, 343–349. https://doi.org/10.1007/978-3-030-68017-6_51

LAMPIRAN



<https://drive.google.com/drive/folders/1kkqP5W4EwhqcEGl6l9pRElbUoBenup8e?usp=sharing>