

***SMART TRAFFIC SYSTEM* UNTUK DETEKSI DAN  
PEMANTAUAN KEMACETAN *REAL-TIME* BERBASIS CCTV  
DI KOTA BANDA ACEH**

**TUGAS AKHIR**

**Diajukan oleh:**

**AMMAR NAUFAL**

**220705093**

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
2026M/1447H**

**LEMBAR PERSETUJUAN**  
**SMART TRAFFIC SYSTEM UNTUK DETEKSI DAN PEMANTAUAN**  
**KEMACETAN REAL-TIME BERBASIS CCTV DI KOTA BANDA ACEH**

**TUGAS AKHIR**

Diajukan kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)  
dalam Program Studi Teknologi Informasi

**Oleh:**

**AMMAR NAUFAL**

**NIM. 220705093**

**Mahasiswa Fakultas Sains dan Teknologi**  
**Program Studi Teknologi Informasi**

**Disetujui Untuk Dimunaqasyahkan Oleh:**

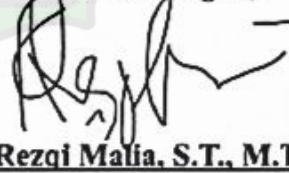
**Pembimbing I,**



**Malahayati, M.T.**

**NIP. 198301272015032003**

**Pembimbing II,**



**Rezqi Malia, S.T., M.T.**

**NIP. 199110262022032004**

Mengetahui,

**Ketua Program Studi Teknologi Informasi**



**Malahayati, M.T.**

**NIP. 198301272015032003**

## LEMBAR PENGESAHAN

### **SMART TRAFFIC SYSTEM UNTUK DETEKSI DAN PEMANTAUAN KEMACETAN *REAL-TIME* BERBASIS CCTV DI KOTA BANDA ACEH**

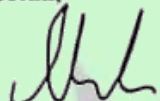
#### TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir  
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus Serta  
Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)  
Dalam Program Studi Teknologi Informasi

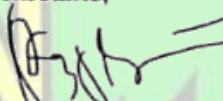
Pada Hari/Tanggal: Selasa, 03 Februari 2026  
15 Sya'ban 1447 H  
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:


Ketua,

  
Malahavati, M.T  
NIP. 198301272015032003


Sekretaris,

  
Rezqi Malia, S.T., M.T.  
NIP. 199110262022032004

Penguji I,

  
Dr. Hendri Ahmadian, M.I.M  
NIP. 198301042014031002

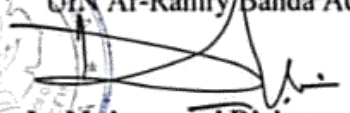
Penguji II,

  
Khairan AR, M.Kom  
NIP. 198607042014031001

Mengetahui,

Dekan Fakultas Sains dan Teknologi  
UIN Ar-Raniry Banda Aceh,



  
Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU  
NIP. 196210021988111001

## LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ammar Naufal  
NIM : 220705093  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul : *Smart traffic system* untuk deteksi dan pemantauan kemacetan *real-time* berbasis cctv di kota banda acch

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggung jawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 15 January

Yang menyatakan



Ammar Naufal

## ABSTRAK

Nama : Ammar Naufal  
NIM : 220705098  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi (FST)  
Judul : *Smart Traffic System* untuk Deteksi dan Pemantauan Kemacetan *Real-Time* Berbasis CCTV di Kota Banda Aceh  
Tanggal Sidang : 29 Januari 2026 / 10 Rajab 1447 H  
Jumlah Halaman : 55 Halaman  
Pembimbing I : Malahayati, M.T.  
Pembimbing II : Rezqi Malia, S.T., M.T.

Kemacetan di Simpang Dharma, Banda Aceh, memerlukan sistem pemantauan otomatis yang cepat dan akurat. Penelitian ini membangun sistem Smart Traffic berbasis web menggunakan model Waterfall dan algoritma YOLOv8n. Model dilatih dengan dataset publik dari Kaggle untuk generalisasi fitur, lalu diimplementasikan pada video CCTV Simpang Dharma melalui optimasi ROI poligon. Hasil pengujian menunjukkan akurasi mean Average Precision (mAP50) sebesar 97,5% dengan kecepatan proses 25-30 FPS. Sistem ini berhasil menyajikan statistik kendaraan dan klasifikasi kepadatan (Lancar, Padat, Macet) secara real-time melalui dashboard website sebagai solusi manajemen lalu lintas digital di Kota Banda Aceh.

**Kata Kunci:** *Smart Traffic*, YOLOv8n, Banda Aceh, *Computer Vision*, *Real-time Detection*.

## ABSTRACT

Name : Ammar Naufal  
Student ID : 220705098  
Major : Teknologi Informasi  
Faculty : Sains dan Teknologi (FST)  
Title : *Smart Traffic System For Real-Time Congestion Detection And Monitoring Based On Cctv In Banda Aceh City*  
Date : 29 January 2026 / 10 Rajab 1447 H  
Total Pages : 55 Pages  
Supervisor I : Malahayati, M.T.  
Supervisor II : Rezqi Malia, S.T., M.T.

Traffic congestion at Simpang Dharma, Banda Aceh, requires a fast and accurate automatic monitoring system. This research develops a web-based Smart Traffic system using the Waterfall model and YOLOv8n algorithm. The model was trained with Kaggle datasets for feature generalization and implemented on Simpang Dharma's CCTV video with polygon ROI optimization. Results show a mean Average Precision (mAP50) of 97.5% at 25-30 FPS. The system successfully provides real-time vehicle statistics and density classification (Smooth, Crowded, Congested) via a web dashboard as a digital traffic management solution for Banda Aceh.

**Keywords:** *Smart Traffic, YOLOv8n, Banda Aceh, Computer Vision, Real-time Detection.*

## KATA PENGANTAR

Segala puji dan rasa syukur penulis haturkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan Proposal Tugas Akhir berjudul “*Smart Traffic System* untuk Deteksi dan Pemantauan Kemacetan *Real-Time* Berbasis CCTV di Kota Banda Aceh” dengan baik dan lancar.

Proposal ini disusun sebagai salah satu syarat dalam menyelesaikan mata kuliah Skripsi pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry Banda Aceh.

Penulis menyadari bahwa proposal ini masih jauh dari kesempurnaan, mengingat keterbatasan pengalaman dan kemampuan yang dimiliki. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang membangun demi penyempurnaan laporan ini di masa mendatang.

Ucapan terima kasih yang mendalam penulis sampaikan kepada semua pihak yang telah memberikan dukungan, bantuan, serta kontribusinya selama proses pelaksanaan dan penyusunan laporan ini yaitu:

1. Orang tua, keluarga, dan adik-adik tercinta atas dukungan, motivasi, serta pengertian yang diberikan selama proses penyusunan skripsi ini. Kehadiran dan doa yang tulus menjadi salah satu hal yang membantu saya menyelesaikan penelitian ini dengan baik.
2. Ibu Malahayati, M. T., selaku Ketua Program Studi Teknologi Informasi sekaligus dosen pembimbing yang telah memberikan bimbingan, masukan, dan arahan selama penyusunan laporan ini.
3. Bu Cut Ida Rahmadiana, S. Si., selaku Staf dan pegawai di Program Studi Teknologi Informasi yang telah memberikan kesempatan, bimbingan, dan kerja sama yang baik.
4. Ibu Rezqi Malia, S.T., M.T., selaku Dosen Pembimbing yang telah dengan penuh kesabaran meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, arahan teknis, serta motivasi yang sangat

berharga bagi penulis sehingga tugas akhir ini dapat terselesaikan dengan baik.

5. Bapak Khairan AR, M.Kom., selaku Dosen Penguji yang telah memberikan kritik, saran, dan masukan konstruktif demi kesempurnaan penelitian ini.
6. Bapak Dr. Hendri Ahmadian, S.Si., M.IM., selaku Dosen Penguji yang telah memberikan arahan serta wawasan mendalam guna meningkatkan kualitas hasil penelitian ini.
7. Rekan-rekan dan teman sekaligus sahabat saya: Siti Tsabita, Nadya Putri, Lady Dwi Ulfa, Fathurrahman, Imamul Muttaqin Mukhvi, Arif Maulana, Muammar Zaki, Muhammad Hafiz, Khairunnisak, Mirzan Maulidar.
8. Kepada Boy's Yala Sulthan Amirul Abrar, Farras Nabil Rivanza, Kausar Rusrijal, Muhammad Ikbar Siddiq yang telah menemani penulis selama di Thailand.
9. Seluruh teman-teman angkatan 2022 Program Studi Teknologi Informasi yang senantiasa memberikan dukungan dan motivasi selama pembuatan proposal ini.

Sebagai penutup, penulis menaruh harapan semoga laporan ini dapat memberi nilai guna serta menjadi sumber rujukan yang berguna bagi siapa pun yang membacanya. Penulis juga berdoa agar segala bentuk bantuan, dukungan, dan kebaikan dari semua pihak memperoleh balasan yang setimpal dari Allah SWT.

## DAFTAR ISI

LEMBAR PERSETUJUAN .....	i
LEMBAR PENGESAHAN .....	ii
LEMBAR KEASLIAN .....	iii
ABSTRAK .....	iv
ABSTRAK .....	iv
ABSTRACT .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL .....	xi
DAFTAR GAMBAR .....	xii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 Penelitian Terdahulu .....	5
2.2 Landasan Teori .....	7
2.2.1 Kemacetan Lalu Lintas.....	7
2.2.2 Sistem Pemantauan Lalu Lintas Berbasis CCTV.....	7
2.2.3 Computer Vision dan Deep Learning.....	8
2.2.4 Algoritma YOLOv8.....	8
2.2.5 Mekanisme <i>Anchor-Free</i> .....	9
2.2.6 Algoritma CSPDarknet ( <i>Cross Stage Partial Darknet</i> ) .....	10
2.3 OpenCV.....	11
2.4 Konsep <i>Smart City</i> .....	12
2.5 Kerangka Pemikiran .....	12
<b>BAB III METODE PENELITIAN .....</b>	<b>14</b>
3.1 Tahapan Penelitian.....	14

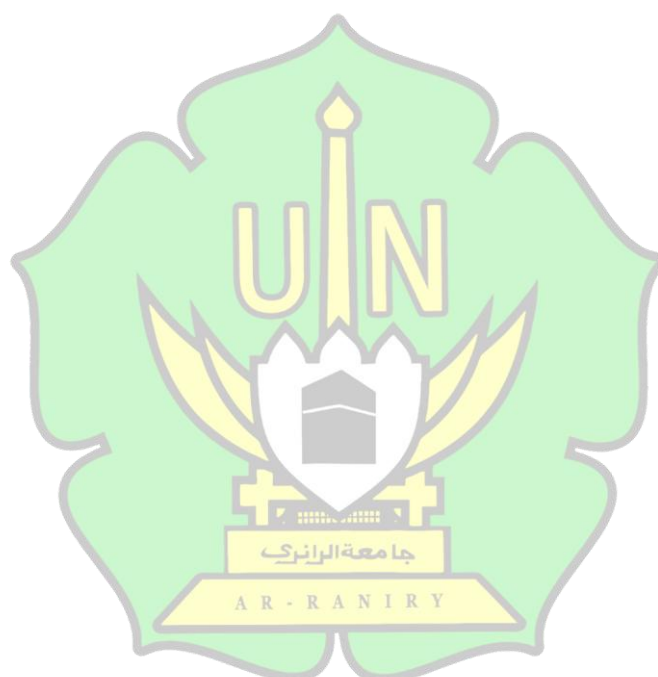
3.1.1 Tahap Analisis Kebutuhan ( <i>Requirements Analysis</i> ).....	15
3.1.2 Tahap Analisis dan Pengumpulan Data ( <i>Analysis and Data Collection</i> ).....	15
3.1.3 Tahap Perancangan Sistem ( <i>Design</i> ).....	15
3.1.4 Tahap Implementasi ( <i>Implementation</i> ).....	16
3.1.5 Tahap Pengujian dan Evaluasi ( <i>Testing and Evaluation</i> ).....	16
3.2 Perancangan Logika Algoritma (Pseudocode) .....	17
3.2.1 Penjelasan Logika Algoritma Pseudocode .....	18
3.3 Jenis dan Pendekatan Penelitian.....	18
3.4 Alat dan Bahan .....	19
3.4.1 Perangkat Keras (Hardware) .....	19
3.4.2 Perangkat Lunak (Software).....	19
3.4.3 Data Penelitian .....	19
3.5 Metode Pengumpulan Data .....	19
3.6 Metode Pengembangan Sistem .....	20
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>22</b>
4.1 Implementasi Perangkat Keras dan Perangkat Lunak.....	22
4.1.1 Lingkungan Pengembangan ( <i>Development Environment</i> ).....	22
4.1.2 Konfigurasi <i>Region of Interest</i> (ROI).....	22
4.2 Hasil Deteksi Kendaraan Menggunakan YOLOv8 .....	23
4.2.1 Analisis Hasil Pelatihan Model Menggunakan Dataset Publik (Kaggle).....	23
4.2.2 Klasifikasi Kendaraan dan Kepercayaan Model .....	28
4.2.3 Analisis Akurasi Deteksi .....	28
4.2.4 Analisis Konvergensi <i>Box Loss</i> dan <i>Distributional Focal Loss</i> (DFL) .....	29
4.2.5 Analisis <i>classification loss</i> ( <i>class loss</i> ) .....	30
4.2.6 Analisis Perbandingan <i>Training</i> dan <i>Validation Loss</i> .....	31
4.2.7 Pengujian Performa Kecepatan Sistem (Frames Per Second)...	32
4.2.8 Analisis Kinerja Model.....	32
4.3 Implementasi dan Analisis Dashboard Website .....	37
4.3.1 Analisis Menu Pemantauan <i>Real-Time</i> .....	37
4.3.2 Analisis Menu Pemantauan Statis .....	38

4.4 Integrasi <i>Backend</i> dan <i>Frontend</i> .....	39
4.5 Pembahasan.....	39
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>40</b>
5.1 Kesimpulan.....	40
5.2 Saran.....	41
<b>DAFTAR PUSTAKA.....</b>	<b>42</b>



## DAFTAR TABEL

Tabel 2.1. 1 Penelitian terdahulu.....	5
--	---



## DAFTAR GAMBAR

<b>Gambar 2.2.4.1</b> Arsitektur YOLOv8 .....	9
Gambar 2.5. 1 Kerangka Pemikiran.....	13
Gambar 3.1. 1 Flowchart Penelitian.....	14
Gambar 3.2. 1 Pseudocode Program .....	17
Gambar 4.2.1.1 Dataset gambar sebelum di-train.....	24
Gambar 4.2.1.2 Pre-trained oleh COCO Dataset .....	24
Gambar 4.2.1.3 Validation set .....	25
Gambar 4.2.1.4 Dataset sudah dilatih dengan COCO Dataset.....	26
Gambar 4.2.1.5 Hasil training model .....	27
Gambar 4.2.4.1 Kurva Box Loss.....	29
Gambar 4.2.4.2 Kurva Distributional Focal Loss .....	30
Gambar 4.2.5 1 Kurva Classification Loss .....	31
Gambar 4.2.8. 1 confusion matrix.....	33
Gambar 4.2.8. 2 Kurva Performa Model.....	34
Gambar 4.2.8. 3 Tabel Ringkasan Performa Model.....	35
<b>Gambar 4.3.1.1</b> Tampilan Dashboard Real-Time .....	37
<b>Gambar 4.3.2.1</b> Menu Pemantauan Statis.....	38





# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kemacetan lalu lintas merupakan permasalahan serius di berbagai kota besar Indonesia, termasuk Kota Banda Aceh. Pertumbuhan jumlah kendaraan yang tidak seimbang dengan kapasitas jalan menyebabkan penumpukan arus lalu lintas yang berdampak pada pemborosan waktu, peningkatan konsumsi bahan bakar, polusi udara, dan kerugian ekonomi yang signifikan. Kementerian Perhubungan (2021) melaporkan bahwa kerugian akibat kemacetan di wilayah perkotaan mencapai triliunan rupiah per tahun, menunjukkan pentingnya inovasi teknologi dalam pengelolaan lalu lintas perkotaan.

Selama ini, sistem pemantauan lalu lintas di Indonesia masih didominasi oleh pengawasan manual menggunakan CCTV dan laporan petugas di lapangan. Metode ini tidak efisien karena informasi kondisi lalu lintas tidak dapat disajikan secara cepat dan real-time kepada masyarakat (Hidayatullah et al., 2025). Dengan adanya kemajuan teknologi *computer vision* dan *deep learning* analisis citra CCTV kini dapat dilakukan secara otomatis, hal ini berpotensi besar meningkatkan keamanan, pemantauan, dan efektivitas investigasi kriminal melalui informasi citra yang lebih baik dan lebih berkualitas.

Salah satu algoritma *deep learning* yang populer dan terbukti efektif dalam mendeteksi objek adalah YOLOv8 (*You Only Look Once*). (Surya & Wahyuni, 2025) menunjukkan bahwa YOLOv8 mampu mendeteksi kendaraan dengan akurasi mencapai 89,4% pada data CCTV Kota Malang. Hasil serupa juga ditemukan oleh (Setiaji et al., 2025) yang memanfaatkan kombinasi YOLOv8 dan OpenCV untuk memantau kepadatan lalu lintas di Kabupaten Demak secara *real-time*. Penelitian (Indra Bayu Pangestu et al., 2024) di Kota Bekasi memperkuat temuan ini dengan menunjukkan kemampuan YOLOv8 dalam bekerja di kondisi nyata meskipun akurasi masih perlu ditingkatkan.

Kota Banda Aceh telah memiliki sejumlah CCTV di persimpangan strategis seperti Simpang Lima dan Simpang Jambo Tape, namun data visual tersebut belum dimanfaatkan secara optimal. Dengan menggabungkan YOLOv8 serta memanfaatkan *COCO Dataset* sebagai data pelatihan untuk meningkatkan kemampuan deteksi, sistem ini diharapkan dapat memberikan informasi kemacetan secara cepat, akurat, dan *real-time* kepada masyarakat.

Berdasarkan permasalahan tersebut, penelitian ini berfokus pada “*Smart Traffic System* untuk Deteksi dan Pemantauan Kemacetan *Real-Time* Berbasis CCTV di Kota Banda Aceh” guna membantu masyarakat memperoleh informasi lalu lintas yang mudah diakses serta mendukung pemerintah dalam mewujudkan sistem transportasi cerdas dan efisien.

## 1.2 Rumusan Masalah

Dengan mempertimbangkan uraian pada bagian latar belakang, maka dapat dirumuskan permasalahan dalam penelitian ini sebagai berikut

1. Bagaimana cara mendeteksi tingkat kemacetan lalu lintas secara *real-time* menggunakan kamera CCTV berbasis algoritma YOLOv8?
2. Bagaimana merancang integrasi antara sistem deteksi lalu lintas dengan *website* agar pengguna jalan dapat mengetahui kondisi jalan kepada pengguna secara otomatis?

## 1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mengembangkan sistem deteksi kemacetan lalu lintas secara *real-time* menggunakan algoritma YOLOv8 berbasis video CCTV.
2. Menyediakan platform informasi lalu lintas yang dapat diakses pengguna melalui *browser* untuk memantau kepadatan kendaraan secara *real-time* maupun berbasis analisis video statis.

## 1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

### 1. Manfaat Akademis

Menjadi referensi ilmiah dalam pengembangan teknologi berbasis *Computer Vision* dan *Deep Learning*, khususnya penerapan YOLOv8 untuk sistem transportasi cerdas di Indonesia.

### 2. Manfaat Praktis

Membantu masyarakat memperoleh informasi kondisi lalu lintas secara cepat dan akurat, serta mendukung pengambilan keputusan perjalanan yang lebih efisien.

### 3. Manfaat Bagi Pemerintah Daerah

Memberikan solusi berbasis teknologi untuk meningkatkan efektivitas sistem pemantauan lalu lintas di Kota Banda Aceh dan mendukung implementasi konsep *Smart City*.

## 1.5 Batasan Masalah

Agar penelitian ini lebih terarah dan fokus pada tujuan yang ingin dicapai, maka ruang lingkup penelitian dibatasi pada beberapa hal berikut:

1. Data yang digunakan berasal dari beberapa video CCTV Kota Banda Aceh. Data berupa video statis (*offline*) maupun streaming (*real-time*) yang menampilkan arus kendaraan.
2. Model deteksi dilatih menggunakan *COCO Dataset* yang telah disesuaikan untuk kendaraan.
3. Penelitian ini berfokus pada aspek deteksi dan pengiriman informasi, bukan pada pengendalian lalu lintas.
4. Indikator Kemacetan  
Tingkat kemacetan ditentukan berdasarkan jumlah kendaraan yang

terdeteksi dalam area pantauan CCTV. Sistem tidak menganalisis kecepatan kendaraan, waktu tempuh, atau kondisi jalan secara fisik, melainkan hanya berdasarkan perbandingan jumlah kendaraan terhadap kapasitas tampilan video.

5. Cakupan Wilayah dan Waktu Penelitian

Studi kasus terbatas pada wilayah Kota Banda Aceh khususnya di Simpang Dharma Banda Aceh, dengan pengujian dilakukan pada periode waktu tertentu yang merepresentasikan jam sibuk (Pagi dan Sore hari).



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

Kajian penelitian sebelumnya berperan penting sebagai acuan dalam memperkokoh landasan teori serta mengidentifikasi ruang penelitian yang masih jarang dieksplorasi. Beberapa studi yang memiliki keterkaitan dengan topik dan metode penelitian ini disajikan pada tabel 2.1.

Tabel 2.1. Penelitian Terdahulu

No	Nama Penulis	Tahun	Judul Penelitian	Metode	Hasil Penelitian
1	Surya & Wahyuni	2025	Deteksi Kendaraan pada CCTV Kota Malang Menggunakan YOLOv8	YOLOv8 + Python	Akurasi deteksi 89,4% dan mampu bekerja secara real-time.
2	Setiaji et al.	2025	Penerapan YOLOv8 dan OpenCV untuk Pemantauan Volume Lalu Lintas	YOLOv8 + OpenCV	Sistem dapat menghitung volume kendaraan dengan error <10%.
3	Nadiyah et al.	2025	Sistem Analisis Kepadatan Lalu Lintas menggunakan Teknologi YOLOv8	YOLOv8	Sistem ini mampu mencapai akurasi deteksi sebesar 88%
4	Rizky Hidayatullah et al.	2025	<i>Deep Learning</i> Algoritma YOLOv8 untuk Meningkatkan Analisis Kepadatan Lalu Lintas.	YOLOv8	Akurasi deteksi tinggi, dengan precision sebesar 0.930, recall 0.919, F1-score 0.930, mAP50 0.975, dan mAP50-95 0.748.
5	(Setiaji et al., 2025)	2025	<i>Real-Time Traffic Density and Anomaly Monitoring Using YOLOv8, OpenCV and Pattern</i>	YOLOv8 + OpenCV + Pattern	Sistem mampu menghitung volume kendaraan dengan error di

			<i>Recognition for Smart City Applications in Demak</i>	<i>Recognition</i>	bawah 10% dan mendeteksi anomali (seperti kendaraan lawan arus/berhenti) dengan akurasi 83,33%.
6	Pangestu et al.	2024	Implementasi YOLOv8 pada Data CCTV Kota Bekasi	YOLOv8	Akurasi 59,2% pada kondisi pencahayaan rendah.
7	(Alo et al., n.d.)	2024	<i>Vehicle Speed Detection System Utilizing YOLOv8 and Sort Algorithm</i>	YOLOv8 + Algoritma SORT + Haversine Formula	Integrasi YOLOv8 dengan algoritma pelacakan SORT mampu mengestimasi kecepatan kendaraan secara <i>real-time</i> dengan tingkat kesalahan ( <i>Mean Absolute Percentage Error</i> ) yang rendah.
8	(Neamah & Karim, 2023)	2023	<i>Real-time Traffic Monitoring System based on Deep Learning Object Detection</i>	YOLOv8 + Custom Dataset	Sistem ini menunjukkan performa tinggi dalam klasifikasi jenis kendaraan (mobil, motor, bus, truk, van) dengan nilai presisi, <i>recall</i> , dan mAP di atas 90%

Berdasarkan table 2.1, dapat disimpulkan bahwa algoritma YOLOv8 menjadi pilihan utama dalam sistem pemantauan *real-time* karena efisiensinya dalam menyeimbangkan kecepatan (*Frame Per Second*) dan akurasi (*Mean Average Precision*). Namun, sebagian besar penelitian tersebut masih berfokus pada kota-kota besar di Jawa. Penelitian ini mengambil celah dengan

mengimplementasikan teknologi tersebut pada infrastruktur spesifik di Kota Banda Aceh, khususnya pada beberapa ruas jalan di Kota Banda Aceh yang dilengkapi dengan CCTV, dengan integrasi langsung ke *platform* berbasis web untuk aksesibilitas publik.

## **2.2 Landasan Teori**

### **2.2.1 Kemacetan Lalu Lintas**

Kemacetan lalu lintas terjadi ketika jumlah kendaraan yang melintasi suatu jalan melebihi daya tampung jalan tersebut, sehingga kecepatan kendaraan menurun dan menimbulkan antrean yang panjang. Beberapa faktor yang memicu kemacetan di antaranya adalah peningkatan jumlah kendaraan bermotor yang tidak sebanding dengan perkembangan infrastruktur jalan, serta pengelolaan lalu lintas yang belum optimal (Mana, 2021).

Menurut Kementerian Perhubungan (2021), kemacetan menimbulkan dampak sosial dan ekonomi yang besar, seperti peningkatan konsumsi bahan bakar, penurunan produktivitas, serta pencemaran udara. Oleh karena itu, diperlukan solusi berbasis teknologi untuk memantau dan mengelola arus lalu lintas secara lebih efektif.

### **2.2.2 Sistem Pemantauan Lalu Lintas Berbasis CCTV**

*Closed Circuit Television* (CCTV) merupakan teknologi pengawasan visual yang umum digunakan untuk memantau kondisi lalu lintas di jalan raya. Data video dari CCTV biasanya diamati secara manual oleh operator untuk menilai kepadatan dan mendeteksi insiden di lapangan. Namun, metode ini memiliki keterbatasan karena bersifat subjektif dan memerlukan tenaga manusia yang banyak (Khan & Shankar, 2024)

Seiring berkembangnya kemajuan teknologi, efektivitas sistem pemantauan dapat ditingkatkan melalui penerapan CCTV cerdas yang terintegrasi dengan teknologi pengolahan citra dan kecerdasan buatan (*Artificial Intelligence* /AI). Sistem tersebut tidak hanya berfungsi merekam tampilan visual, tetapi juga memiliki kemampuan untuk menganalisis gambar maupun video secara otomatis guna mengidentifikasi kejadian yang berpotensi menimbulkan bahaya, seperti

kecelakaan lalu lintas, kemacetan, atau kendaraan yang berhenti di area berisiko (Bintan & Asahan, 2025).

### 2.2.3 Computer Vision dan Deep Learning

*Computer Vision* adalah bidang ilmu yang memungkinkan komputer menafsirkan dan memahami gambar atau video secara otomatis. Dalam konteks lalu lintas, teknologi ini dapat digunakan untuk mendeteksi kendaraan, menghitung jumlahnya, hingga menganalisis tingkat kepadatan jalan.

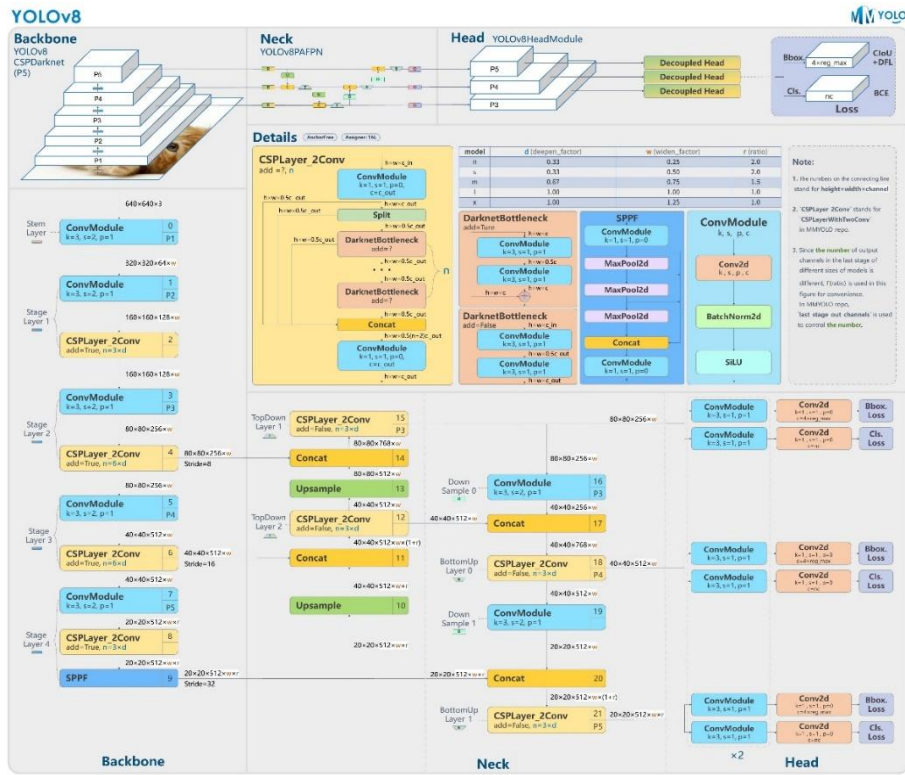
Teknologi ini semakin berkembang dengan adanya *Deep Learning*, terutama melalui CNN (*Convolutional Neural Network*), yang mampu mengenali objek dengan akurasi tinggi. CNN bekerja dengan mengekstraksi fitur-fitur penting dari gambar, seperti bentuk dan pola, yang kemudian digunakan untuk mengenali objek tertentu.

### 2.2.4 Algoritma YOLOv8

YOLO (*You Only Look Once*) merupakan salah satu algoritma object detection yang dirancang untuk mendeteksi dan melokalisasi objek dalam satu kali proses komputasi. Versi terbarunya, YOLOv8, dikembangkan oleh Ultralytics pada tahun 2023 dan dikenal memiliki peningkatan performa signifikan dibanding versi sebelumnya (Ultralytics, 2023).

YOLOv8 menggunakan pendekatan *anchor-free detection* dengan arsitektur CSPDarknet53 yang disempurnakan, sehingga dapat mencapai keseimbangan antara kecepatan dan akurasi. Dalam konteks lalu lintas, YOLOv8 mampu mendeteksi berbagai jenis kendaraan seperti mobil, motor, dan bus dalam kondisi pencahayaan dan cuaca yang bervariasi.

Penelitian oleh (Surya & Wahyuni, 2025) menunjukkan bahwa YOLOv8 memiliki tingkat akurasi deteksi kendaraan sebesar 89,4% pada video CCTV di Kota Malang. Sementara (Setiaji et al., 2025) mengintegrasikan YOLOv8 dengan OpenCV untuk menghitung volume lalu lintas, dan hasilnya menunjukkan performa yang baik untuk sistem *real-time*.



Gambar 2.2.4.1 Arsitektur YOLOv8

Arsitektur YOLOv8 terdiri dari tiga bagian utama, yaitu backbone, neck, dan head. Bagian backbone berfungsi untuk mengekstraksi fitur dari citra masukan melalui serangkaian proses konvolusi dan *feature extraction* bertingkat. Selanjutnya, bagian *neck* berperan dalam menggabungkan dan memperkaya fitur dari berbagai skala menggunakan mekanisme *feature pyramid*, sehingga informasi objek dengan ukuran berbeda dapat dipertahankan. Adapun bagian head digunakan untuk melakukan proses prediksi, yang meliputi klasifikasi objek, penentuan *bounding box*, dan estimasi tingkat kepercayaan (*confidence score*) secara langsung dalam satu tahap (*one-stage detector*). Dengan arsitektur tersebut, YOLOv8 mampu melakukan deteksi objek secara cepat dan akurat.

### 2.2.5 Mekanisme Anchor-Free

Mekanisme *Anchor-Free* merupakan paradigma baru dalam deteksi objek yang menghilangkan ketergantungan pada *anchor boxes* (kotak acuan yang telah ditentukan ukurannya sebelumnya). Pada algoritma deteksi tradisional seperti YOLO versi awal hingga YOLOv5, sistem menggunakan sekumpulan kotak acuan

dengan berbagai rasio aspek dan ukuran yang harus didefinisikan secara manual sebelum proses pelatihan dimulai (Terven, 2023)

Dalam pendekatan *Anchor-Free*, model tidak lagi mencoba mencocokkan objek dengan kotak-kotak acuan tersebut. Sebagai gantinya, sistem secara langsung memprediksi pusat dari sebuah objek (*object center*) dan jarak dari pusat tersebut ke empat sisi pembatasnya (atas, bawah, kiri, dan kanan). Hal ini menyederhanakan proses deteksi karena model memperlakukan setiap lokasi pada feature map sebagai potensi pusat objek.

Keunggulan utama dari mekanisme ini adalah fleksibilitas yang lebih tinggi dalam mendeteksi objek dengan berbagai skala dan rasio aspek yang ekstrem, yang sering kali sulit ditangkap secara akurat oleh *anchor boxes* yang kaku. Selain itu, penghilangan anchor secara signifikan mengurangi jumlah parameter dan komputasi yang dibutuhkan, serta meminimalkan kebutuhan akan penyetelan hiperparameter (*hyperparameter tuning*) yang kompleks, sehingga proses pelatihan menjadi lebih stabil dan cepat.

#### **2.2.6 Algoritma CSPDarknet (*Cross Stage Partial Darknet*)**

CSPDarknet adalah arsitektur tulang punggung (*backbone*) yang dirancang untuk meningkatkan efisiensi pembelajaran pada jaringan syaraf dalam. Arsitektur ini merupakan pengembangan dari Darknet konvensional dengan mengintegrasikan konsep *Cross Stage Partial Network (CSPNet)*.

Logika utama di balik CSPDarknet adalah pemisahan aliran gradien. Pada setiap blok tingkatan jaringan, peta fitur (*feature map*) dari lapisan dasar dibagi menjadi dua bagian. Bagian pertama melewati blok konvolusi yang intensif secara komputasi, sementara bagian kedua melewati jalur pintas (*bypass*) untuk langsung digabungkan kembali di akhir tahap. Pemisahan ini bertujuan untuk mengurangi informasi gradien yang berulang serta meminimalkan duplikasi data selama proses propagasi balik (*back propagation*).

Penerapan algoritma ini memberikan beberapa keuntungan teknis yang signifikan:

- Efisiensi Komputasi: Dengan mengurangi redundansi pada peta fitur, CSPDarknet mampu mempertahankan akurasi yang tinggi meskipun menggunakan jumlah operasi *floating-point* (FLOPs) yang lebih sedikit.
- Optimalisasi Memori: Struktur ini memungkinkan penggunaan memori yang lebih hemat, sehingga sangat ideal untuk dijalankan pada perangkat dengan sumber daya terbatas.
- Peningkatan Aliran Gradien: Dengan adanya jalur pintas, informasi fitur dari lapisan awal dapat tersampaikan lebih baik ke lapisan yang lebih dalam, yang pada akhirnya mempercepat konvergensi model saat proses pelatihan dan meningkatkan kemampuan ekstraksi fitur visual yang kompleks.
- Integrasi antara mekanisme *Anchor-Free* dan CSPDarknet menciptakan sebuah sistem deteksi yang tidak hanya akurat dalam mengenali objek secara detail, tetapi juga sangat efisien dalam hal kecepatan pemrosesan data citra atau video.

### 2.3 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah pustaka pemrograman *open-source* yang digunakan untuk pemrosesan citra dan video. OpenCV menyediakan berbagai fungsi untuk manipulasi gambar, pengenalan objek, dan pelacakan pergerakan. Dalam penelitian ini, OpenCV digunakan untuk membaca aliran video dari CCTV, menggambar *bounding box* hasil deteksi kendaraan, serta menghitung jumlah kendaraan dalam area tertentu.

Lebih jauh lagi, dalam pengembangan sistem pemantauan cerdas tingkat lanjut, OpenCV berperan vital sebagai jembatan integrasi antara modul deteksi dan algoritma pelacakan (*tracking*). Pustaka ini memfasilitasi mekanisme pelacakan *multi-object* (seperti BoTSORT atau DeepSORT) yang mampu mempertahankan identitas unik (ID) setiap kendaraan antar-*frame* video. Kemampuan ini sangat krusial untuk mengatasi masalah oklusi—di mana kendaraan tertutup oleh kendaraan lain—sehingga sistem dapat memastikan akurasi data volume lalu lintas yang presisi tanpa terjadinya penghitungan ganda, sebagaimana dibuktikan dalam

studi terbaru mengenai optimasi pelacakan kendaraan berbasis YOLOv8 (Shen, 2025).

#### **2.4 Konsep *Smart City***

Konsep *Smart City* merupakan pendekatan pengelolaan kota yang mengintegrasikan teknologi informasi untuk meningkatkan efisiensi pelayanan publik. Salah satu aspek pentingnya adalah *Smart Mobility*, yaitu Sistem yang dirancang untuk mengurangi kemacetan lalu lintas, meningkatkan aksesibilitas, dan mempermudah mobilitas masyarakat di dalam kota. Penerapan sistem deteksi kemacetan berbasis AI di Banda Aceh mendukung terwujudnya *Smart Mobility* sebagai bagian dari *Smart City* (Oktaviana & Azkia Asca, 2023).

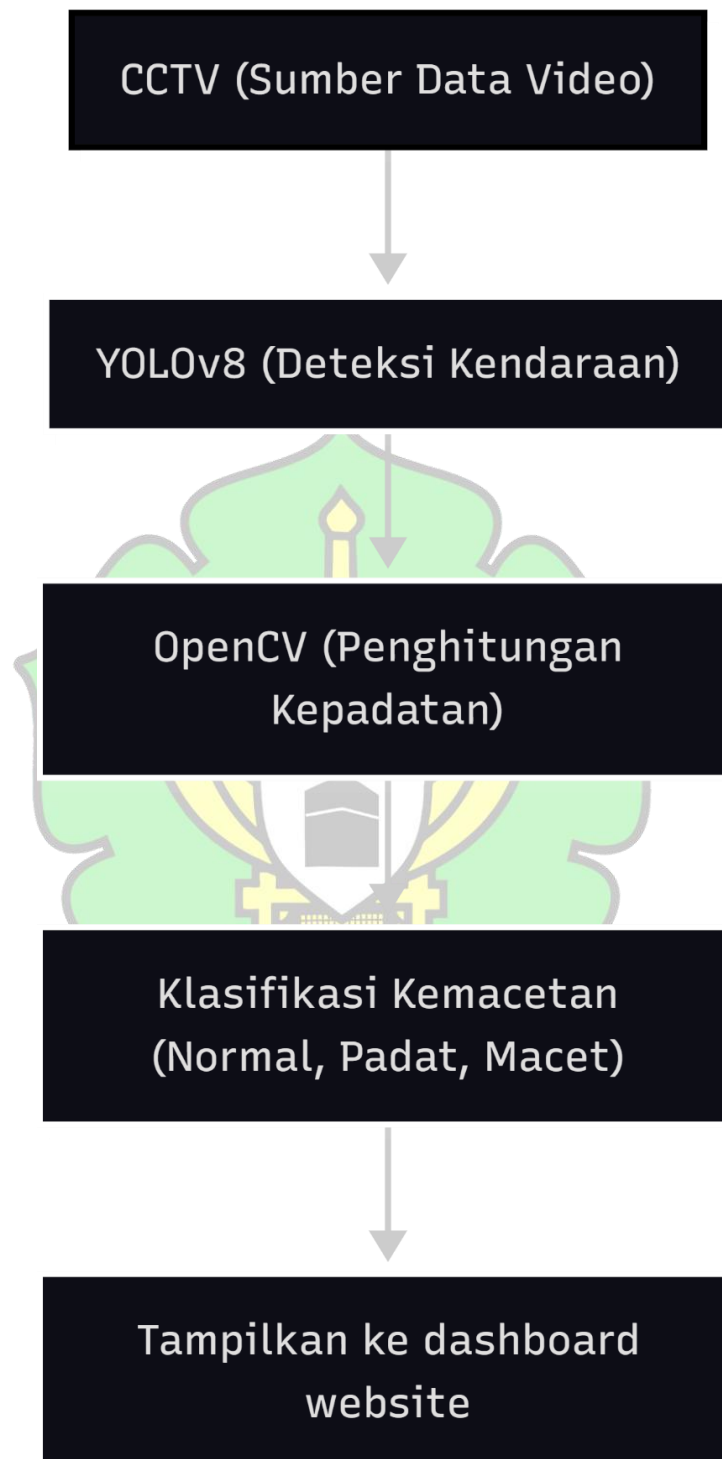
Dalam ekosistem *Smart Mobility*, keandalan data lalu lintas menjadi kunci utama keberhasilan manajemen transportasi kota. Implementasi *Intelligent Transportation Systems* (ITS) yang memanfaatkan algoritma *Deep Learning* mutakhir seperti YOLOv8 terbukti mampu menggantikan metode survei manual yang lambat dan rentan kesalahan. Dengan kemampuan deteksi dan klasifikasi objek yang presisi, sistem ini dapat menyediakan statistik volume kendaraan secara *real-time* kepada pemangku kebijakan, memungkinkan respons cepat terhadap dinamika kemacetan yang terjadi di lapangan (Dharmasaputra & Hartato, 2025).

#### **2.5 Kerangka Pemikiran**

Kerangka berpikir dalam penelitian ini menggambarkan alur logika dari identifikasi masalah hingga pengembangan solusi. Permasalahan kemacetan lalu lintas di Banda Aceh disebabkan oleh keterbatasan sistem pemantauan yang masih manual. Data CCTV yang ada belum diolah secara otomatis untuk mendeteksi kepadatan kendaraan.

Penelitian ini mengusulkan solusi berupa sistem deteksi kemacetan lalu lintas *real-time* dengan memanfaatkan algoritma YOLOv8 untuk mengenali dan menghitung jumlah kendaraan. Hasil deteksi kemudian diklasifikasikan menjadi tingkat kepadatan lalu lintas dan dikirimkan melalui website agar masyarakat dapat segera mengetahui kondisi jalan terkini.

Secara konseptual, kerangka berpikir penelitian ini dapat digambarkan sebagai berikut:



*Gambar 2.5. 1 Kerangka Pemikiran*

## BAB III

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian



Gambar 3.1. 1 Flowchart Penelitian

Diagram alir penelitian pada Gambar 3.1.1 merupakan representasi visual dari tahapan sistematis yang dilakukan dalam membangun Smart Traffic System. Penjelasan lebih rinci mengenai setiap tahapan pada diagram alir tersebut adalah sebagai berikut:

### 3.1.1 Tahap Analisis Kebutuhan (*Requirements Analysis*)

Tahap ini merupakan langkah awal untuk memetakan spesifikasi teknis dan fungsional sistem. Peneliti melakukan identifikasi terhadap kebutuhan perangkat keras yang mampu menangani komputasi *Deep Learning*, seperti GPU untuk proses pelatihan dan CPU dengan instruksi tinggi untuk proses inferensi. Selain itu, dilakukan identifikasi perangkat lunak yang mencakup *environment* Python, pustaka OpenCV, dan arsitektur YOLOv8. Fokus utama pada tahap ini adalah memastikan seluruh instrumen siap untuk menangani data video *real-time* dari CCTV Kota Banda Aceh.

### 3.1.2 Tahap Analisis dan Pengumpulan Data (*Analysis and Data Collection*)

Berbeda dengan penelitian deteksi objek konvensional, tahap ini dibagi menjadi dua fokus utama:

- 1) Akuisisi Dataset Publik (Kaggle): Peneliti mengumpulkan data gambar kendaraan yang telah terlabeli secara standar dari *platform* Kaggle. Penggunaan dataset publik ini bertujuan untuk memberikan variansi data yang luas (pencahayaan, sudut pandang, dan jenis kendaraan) agar model memiliki kemampuan generalisasi yang kuat sebelum dihadapkan pada data nyata.
- 2) Analisis Karakteristik CCTV: Peneliti melakukan observasi terhadap aliran video dari CCTV Kota Banda Aceh untuk menentukan resolusi dan sudut pandang (*angle*) yang akan digunakan sebagai input pada tahap implementasi nantinya.

### 3.1.3 Tahap Perancangan Sistem (*Design*)

Pada tahap ini, peneliti mentransformasikan kebutuhan sistem ke dalam desain teknis.

- 1) Perancangan (*Region Of Ineterest*) ROI Poligon: Dilakukan desain koordinat poligon yang akan membatasi area kerja algoritma. Penentuan poligon ini sangat krusial agar sistem tidak menghitung kendaraan di luar jalur jalan raya (seperti area trotoar), sehingga meningkatkan efisiensi komputasi.
- 2) Desain Dashboard Web: Perancangan antarmuka pengguna dilakukan untuk memastikan informasi statistik kendaraan dan status kepadatan dapat ditampilkan secara informatif dan responsif.

### 3.1.4 Tahap Implementasi (*Implementation*)

Tahap ini merupakan inti dari realisasi penelitian, di mana terdapat dua proses utama:

- 1) Pelatihan Model (*Model Training*): Model YOLOv8 dilatih menggunakan dataset publik yang telah diunduh dari Kaggle. Proses ini menghasilkan file bobot (*weight*) yang menyimpan pengetahuan tentang fitur-fitur kendaraan.
- 2) Penyatuan Sistem (*Deployment*): File bobot tersebut kemudian diimplementasikan untuk melakukan deteksi pada data CCTV Kota Banda Aceh secara *real-time*. Di sini, mesin AI akan memproses video asli Banda Aceh menggunakan "kecerdasan" yang didapat dari dataset Kaggle. Secara bersamaan, *frontend* website dibangun untuk menampilkan hasil olahan data tersebut.

### 3.1.5 Tahap Pengujian dan Evaluasi (*Testing and Evaluation*)

Tahap akhir ini berfungsi sebagai validasi terhadap seluruh rangkaian proses.

- 1) Evaluasi Metrik Akurasi: Performa model diukur menggunakan parameter mAP (*mean Average Precision*). Hasil penelitian menunjukkan nilai mAP50 mencapai 97.5%, yang menandakan model sangat akurat dalam mengenali objek di lapangan.
- 2) Validasi Fungsional: Menguji apakah klasifikasi status kepadatan (Lancar, Padat, Macet) sudah sesuai dengan logika jumlah kendaraan yang ditetapkan. Pengujian dilakukan melalui metode *Black Box* untuk memastikan seluruh fitur dashboard berjalan sukses tanpa kegagalan sistem.

### 3.2 Perancangan Logika Algoritma (Pseudocode)

Untuk memberikan gambaran yang jelas mengenai mekanisme kerja sistem deteksi kemacetan, bagian ini menguraikan alur logika pemrosesan data mulai dari input aliran video hingga penghasilan status kepadatan lalu lintas. Logika ini dirancang untuk beroperasi secara real-time dengan efisiensi tinggi.

```
-----
Pseudocode: Sistem Pemantauan Kepadatan Lalu Lintas Menggunakan YOLOv8
-----
INPUT: Aliran_Video_CCTV (Real-time)
OUTPUT: Status_Kepadatan (Dashboard Web)

DEKLARASI:
  Model = Muat_Bobot_YOLOv8(Kaggle_Weights)
  ROI = Koordinat_Poligon_Jalan
  Threshold_Macet = 25
  Threshold_Padat = 10

MULAI:
  SELAMA (Kamera_Aktif) LAKUKAN:
    Frame = Baca_Bingkai_Video()
    Objek_Terdeteksi = Model.Prediksi(Frame)

    Hitung_Kendaraan = 0

    UNTUK SETIAP Objek DALAM Objek_Terdeteksi LAKUKAN:
      JIKA (Titik_Tengah(Objek) berada_di_dalam ROI) MAKA:
        Hitung_Kendaraan = Hitung_Kendaraan + 1
        Gambarkan_Bounding_Box(Objek)
      AKHIR JIKA
    AKHIR UNTUK

    // Logika Klasifikasi Kepadatan
    JIKA (Hitung_Kendaraan >= Threshold_Macet) MAKA:
      Status = "Macet"
      Warna_Indikator = "Merah"
    ATAU JIKA (Hitung_Kendaraan >= Threshold_Padat) MAKA:
      Status = "Padat"
      Warna_Indikator = "Kuning"
    LAINNYA:
      Status = "Normal/Lancar"
      Warna_Indikator = "Hijau"
    AKHIR JIKA

    Kirim_Data_ke_Dashboard(Hitung_Kendaraan, Status, Warna_Indikator)
  AKHIR SELAMA
SELESAI
-----
```

Gambar 3.2. 1 Pseudocode Program

### 3.2.1 Penjelasan Logika Algoritma Pseudocode

Berdasarkan pseudocode pada gambar 3.2.1, terdapat beberapa tahapan inti yang menjelaskan bagaimana sistem mengolah data:

- Inisialisasi Model dan Parameter: Sistem memulai dengan memuat file bobot (*weights*) YOLOv8 yang telah dilatih menggunakan dataset publik dari Kaggle. Selain itu, didefinisikan variabel *Region of Interest* (ROI) dalam bentuk koordinat poligon serta ambang batas (*threshold*) kendaraan untuk menentukan kategori kepadatan.
- Pemrosesan Bingkai (*Frame Processing*): Sistem melakukan pembacaan bingkai video secara terus-menerus. Setiap bingkai akan dianalisis oleh model YOLOv8 untuk mengidentifikasi keberadaan objek kendaraan seperti mobil, motor, bus, dan truk.
- *Region Of Interest* (ROI): Peneliti menerapkan logika spasial di mana sistem hanya akan menghitung objek yang titik tengahnya (centroid) berada di dalam area ROI poligon. Hal ini bertujuan untuk meningkatkan akurasi data dan mengabaikan kendaraan yang terparkir di bahu jalan atau objek di luar ruas jalan utama.
- Klasifikasi Status Kepadatan: Hasil perhitungan jumlah kendaraan di dalam ROI kemudian dibandingkan dengan nilai ambang batas. Logika thresholding ini akan secara otomatis menetapkan status "Macet" apabila volume kendaraan sangat tinggi, "Padat" untuk volume sedang, dan "Normal/Lancar" untuk volume rendah.
- Output Data: Hasil akhir berupa angka statistik, teks status, dan representasi warna indikator akan dikirimkan secara asinkron ke dashboard website agar pengguna dapat menerima informasi kondisi lalu lintas Kota Banda Aceh secara instan.

### 3.3 Jenis dan Pendekatan Penelitian

Dalam penelitian ini digunakan pendekatan kuantitatif dengan rancangan eksperimen yang menitikberatkan pada proses pengembangan serta pengujian sistem deteksi kepadatan lalu lintas secara *real-time*, menggunakan algoritma YOLOv8 yang diintegrasikan kedalam website.

Pendekatan tersebut dipilih karena penelitian ini melibatkan tahapan pengumpulan data berupa gambar dan video, pelatihan model dalam mengenali objek kendaraan, serta evaluasi performa sistem berdasarkan hasil deteksi dan pengiriman notifikasi otomatis kepada pengguna.

### **3.4 Alat dan Bahan**

#### **3.4.1 Perangkat Keras (Hardware)**

Laptop / PC dengan spesifikasi minimal:

- Laptop / PC dengan spesifikasi minimal:
  - Prosesor: Intel Core i5 / Ryzen 5
  - Ram: 8 GB
  - GPU dengan VRAM 1gb (disarankan untuk pelatihan YOLOv8)
  - Kamera atau rekaman Video CCTV area lalu lintas

#### **3.4.2 Perangkat Lunak (Software)**

- Sistem Operasi: Windows 10 / 11
- Bahasa Pemrograman: Python 3.10
- Library Utama:
  - YOLOv8 (*ultralytics*) untuk deteksi objek
  - OpenCV untuk pemrosesan citra dan video
  - Numpy, Pandas, Matplotlib untuk analisis data
  - Jupyter Notebook / vscode sebagai IDE pengembangan

#### **3.4.3 Data Penelitian**

Data yang digunakan berupa:

1. Video CCTV lalu lintas di Kota Banda Aceh.
2. Dataset kendaraan (mobil, motor, bus, truk) yang digunakan untuk pelatihan atau validasi model YOLOv8 (dapat berasal dari dataset publik seperti coco atau dataset hasil anotasi manual).

### **3.5 Metode Pengumpulan Data**

Metode pengumpulan data dilakukan melalui dua cara utama:

1. Dataset Primer, menggunakan dataset publik yang tersedia di Kaggle
2. Dataset Sekunder, yaitu pengumpulan data citra kendaraan dari sumber terbuka seperti *coco dataset*, *roboflow*, atau dataset penelitian sebelumnya.
3. Observasi langsung, dengan merekam video lalu lintas menggunakan kamera atau mengambil data dari cctv publik.

Seluruh data yang dikumpulkan akan melalui proses *preprocessing*, seperti *cropping area* relevan, *resize* gambar, dan augmentasi data untuk meningkatkan performa model deteksi.

### 3.6 Metode Pengembangan Sistem

Penelitian ini menggunakan metode *Prototyping* karena memungkinkan pengembangan sistem secara bertahap dan interaktif. setiap tahap dapat diuji dan disempurnakan sebelum menuju implementasi akhir.

Tahapan pengembangan terdiri atas:

#### 1. Analisis Kebutuhan

Menentukan kebutuhan sistem, baik dari sisi perangkat keras, perangkat lunak, serta data yang dibutuhkan. pada tahap ini juga dilakukan analisis terhadap format video CCTV dan jenis kendaraan yang akan dideteksi.

#### 2. Perancangan Sistem

Merancang arsitektur sistem yang meliputi alur data, integrasi model YOLOv8, pemrosesan video dengan OpenCV, serta mekanisme pengiriman data ke website.

komponen utama sistem:

- **Input** : Video CCTV/ kamera *real-time*
- **Proses** : Deteksi kendaraan menggunakan YOLOv8
- **Output** : Informasi tingkat kepadatan lalu lintas melalui website

#### 3. Implementasi Sistem

Pada tahap ini dilakukan pengkodean menggunakan Python. YOLOv8 akan diintegrasikan dengan OpenCV untuk membaca dan menganalisis *frame* video, menghitung jumlah kendaraan, serta menentukan tingkat kemacetan berdasarkan ambang batas tertentu.

#### 4. Pengujian dan Evaluasi

Sistem diuji menggunakan video nyata di beberapa lokasi. evaluasi dilakukan terhadap:

- Akurasi deteksi kendaraan
- Ketepatan klasifikasi tingkat kemacetan
- Kecepatan pemrosesan *Frames per Second* (FPS)
- Akurasi informasi di website

#### 5. Training Model

Model YOLOv8n akan dilatih menggunakan dataset publik dari kaggle untuk latihan data awal, versi YOLOv8 yang akan dipakai adalah versi *nano*. YOLOv8n (*nano*) adalah versi paling ringan dari YOLOv8, dengan parameter yang lebih sedikit, model ini mampu memberikan performa dan kecepatan pemrosesan yang lebih efektif daripada model lainnya. Dan juga untuk menghemat sumber daya, versi *nano* dirancang untuk berjalan optimal bahkan pada perangkat dengan spesifikasi menengah.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Perangkat Keras dan Perangkat Lunak

Tahap implementasi merupakan perwujudan dari rancangan sistem yang telah disusun pada bab sebelumnya. Implementasi ini mencakup penyediaan infrastruktur komputasi dan konfigurasi lingkungan pemrograman agar algoritma YOLOv8 dapat berjalan secara optimal untuk mendeteksi kendaraan secara *real-time*.

##### 4.1.1 Lingkungan Pengembangan (*Development Environment*)

Perangkat keras yang digunakan dalam penelitian ini memiliki peran krusial, terutama dalam menjaga kestabilan *Frame Per Second* (FPS) saat proses inferensi video berlangsung. Penggunaan prosesor Intel i7 evo gen 11 dan memori RAM sebesar 16 GB memungkinkan sistem untuk melakukan pemrosesan data citra dari CCTV Kota Banda Aceh tanpa mengalami *bottleneck* yang berarti. Selain itu, jika terdapat dukungan GPU, proses komputasi paralel pada arsitektur YOLOv8 akan menjadi jauh lebih cepat dibandingkan hanya mengandalkan CPU.

Dari sisi perangkat lunak, Python versi 3.10 dipilih sebagai basis bahasa pemrograman karena fleksibilitasnya dalam mengintegrasikan berbagai pustaka kecerdasan buatan. Pustaka *ultralytics* digunakan sebagai kerangka kerja utama untuk menjalankan model YOLOv8. Selain itu, pustaka *OpenCV* berperan dalam manajemen *frame* video, mulai dari pengambilan gambar (*frame grabbing*), konversi warna, hingga pemberian anotasi berupa kotak pembatas (*bounding box*) dan label teks pada objek yang terdeteksi.

##### 4.1.2 Konfigurasi *Region of Interest* (ROI)

Untuk meningkatkan efisiensi komputasi dan akurasi deteksi, sistem ini menerapkan konsep *Region of Interest* (ROI). Pada video *live stream* dari CCTV Kota Banda Aceh khususnya di Simpang Dharma, tidak semua area di dalam *frame* merupakan area jalan raya yang relevan untuk dihitung kepadatannya. Terdapat

objek-objek statis seperti pohon, bangunan, atau trotoar yang jika diproses akan menambah beban kerja sistem secara sia-sia.

Penentuan ROI dilakukan dengan menetapkan koordinat poligon yang mencakup badan jalan di Simpang Dharma. Dengan membatasi area deteksi hanya pada koordinat tersebut, algoritma YOLOv8 hanya akan melakukan pemindaian objek di dalam area jalan. Hal ini terbukti mampu mengurangi kemunculan *false positive* (objek non-kendaraan yang dianggap kendaraan) dan secara signifikan meningkatkan kecepatan pemrosesan data per detik.

## **4.2 Hasil Deteksi Kendaraan Menggunakan YOLOv8**

Hasil deteksi menunjukkan bahwa model YOLOv8n (*nano*) yang diimplementasikan mampu mengenali berbagai kelas kendaraan di lingkungan urban Banda Aceh dengan tingkat kepercayaan (*confidence score*) yang cukup tinggi.

### **4.2.1 Analisis Hasil Pelatihan Model Menggunakan Dataset Publik (Kaggle)**

Sebelum sistem diimplementasikan pada data real-time CCTV Simpang Dharma, peneliti terlebih dahulu melakukan fase pelatihan (*training*) dan pengujian awal menggunakan dataset publik yang bersumber dari platform Kaggle. Penggunaan dataset publik ini bertujuan untuk memberikan basis pengetahuan yang luas kepada model YOLOv8n mengenai berbagai karakteristik visual kendaraan (mobil, motor, bus, dan truk) dalam berbagai kondisi pencahayaan, sudut pandang, dan latar belakang yang beragam.

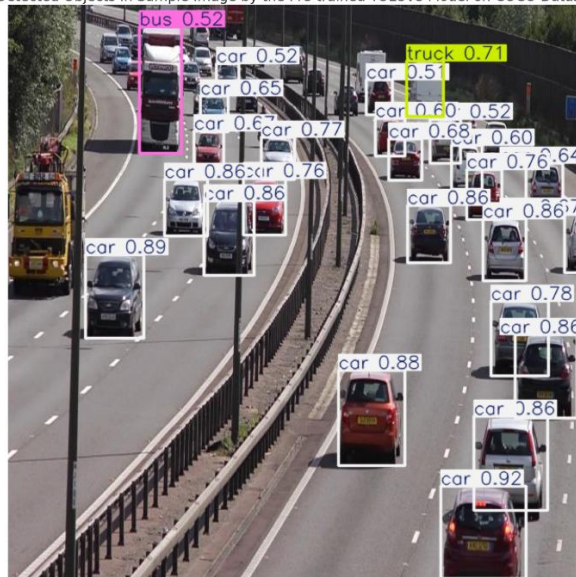
Sample Images from Training Dataset



Gambar 4.2.1.1 Dataset gambar sebelum di-train

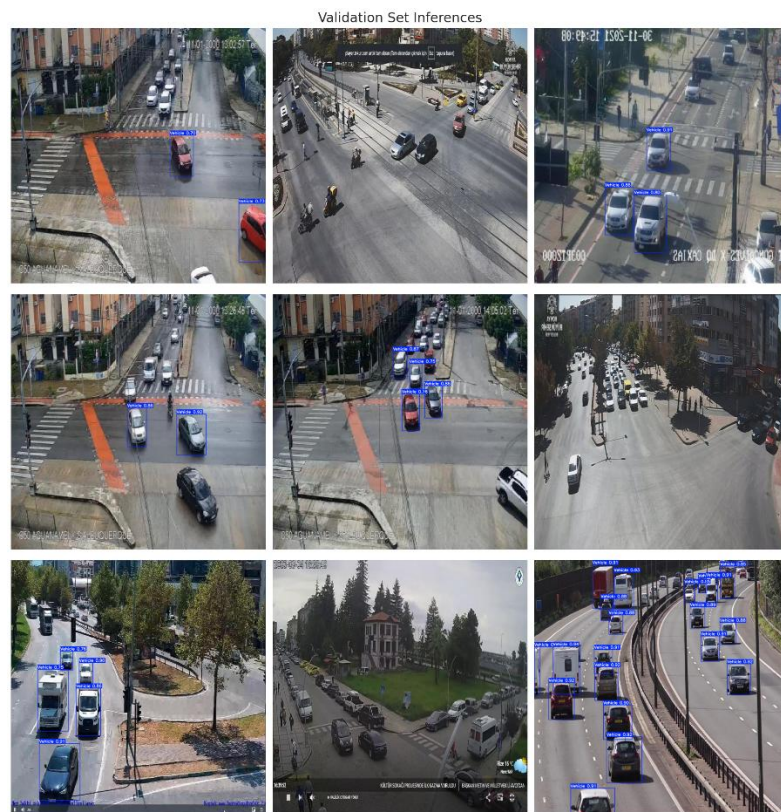
Gambar 4.2.1.1 merepresentasikan data primer yang diperoleh dari Kaggle sebelum dilakukan proses anotasi. Citra ini merupakan kumpulan foto jalan raya dengan berbagai jenis kendaraan dalam kondisi lingkungan yang beragam. Pada tahap ini, gambar belum memiliki informasi spasial (koordinat) yang dapat dibaca oleh algoritma. Fungsi dari visualisasi ini adalah untuk menunjukkan kualitas dan variansi data awal yang akan digunakan sebagai bahan dasar pembelajaran mesin guna membangun kemampuan generalisasi model yang kuat.

Detected Objects in Sample Image by the Pre-trained YOLOv8 Model on COCO Dataset



Gambar 4.2.1.2 Pre-trained oleh COCO Dataset

Gambar 4.2.1.2 merepresentasikan "titik awal" pengetahuan model sebelum dilatih dengan data spesifik kendaraan Anda. Arsitektur YOLOv8n yang digunakan telah dibekali dengan bobot (*weights*) dari COCO Dataset (*Common Objects in Context*). Dataset ini terdiri dari ribuan citra dengan 80 kategori objek umum. Penggunaan *pre-trained* model ini memungkinkan diterapkannya teknik *Transfer Learning*, di mana model tidak belajar dari nol, melainkan memanfaatkan pemahaman dasar tentang bentuk, tepi, dan tekstur yang sudah dipelajari sebelumnya untuk mempercepat proses konvergensi.



Gambar 4.2.1.3 Validation set

Gambar 4.2.1.3 menampilkan kumpulan data dari Kaggle yang sengaja dipisahkan dan tidak digunakan selama proses pelatihan (*training*). Visualisasi pada validation set ini berfungsi sebagai "ujian" bagi model untuk melihat sejauh mana model dapat mengenali objek yang belum pernah dilihat sebelumnya. Jika *bounding box* pada gambar validasi ini terlihat presisi dan labelnya akurat, maka hal tersebut membuktikan bahwa model telah berhasil mempelajari fitur-fitur kendaraan secara universal dan siap untuk diimplementasikan pada kondisi dunia nyata.

Detected Objects in Sample Image by the Fine-tuned YOLOv8 Model



Gambar 4.2.1.4 Dataset sudah dilatih dengan COCO Dataset

Gambar 4.2.1.4 merupakan visualisasi hasil akhir setelah model dilatih selama 100 epoch menggunakan dataset dari Kaggle. Pada gambar ini, terlihat model sudah mampu melakukan prediksi secara mandiri dengan memberikan label (mobil, bus, motor, truk) beserta tingkat kepercayaan (*confidence score*) di atas objek kendaraan. Hasil *fine-tuning* ini mencerminkan pencapaian mAP50 sebesar 97.5%, yang menunjukkan bahwa model telah berhasil mengadaptasi pengetahuan umumnya menjadi spesialisasi deteksi kendaraan untuk kebutuhan sistem Smart Traffic di Simpang Dharma.

	Metric Value
metrics/precision(B)	0.917
metrics/recall(B)	0.929
metrics/mAP50(B)	0.975
metrics/mAP50-95(B)	0.738
fitness	0.738

Gambar 4.2.1.5 Hasil training model

Berdasarkan Gambar 4.2.1.5 yang merangkum hasil evaluasi model YOLOv8n pada validation set dari dataset Kaggle, diperoleh parameter-parameter metrik yang menunjukkan tingkat keberhasilan pelatihan sebagai berikut:

- Analisis mAP50 (Akurasi Utama) Nilai mAP50 sebesar 97.5% merupakan metrik utama yang merepresentasikan akurasi model pada ambang batas *Intersection over Union* (IoU) 0.5. Angka ini merujuk pada kolom *metrics/mAP50(B)* dalam catatan pelatihan. Dalam konteks sistem ini, persentase 97.5% menunjukkan bahwa model memiliki kemampuan yang sangat tinggi dalam mendeteksi dan mengklasifikasikan objek kendaraan secara bersamaan dengan tingkat kesalahan yang sangat minimal.
- Analisis *Precision* (Ketelitian) Merujuk pada nilai 91.7% yang diambil dari kolom *metrics/precision(B)*, angka ini menggambarkan tingkat ketelitian model. Persentase ini membuktikan bahwa dari 100 objek yang diprediksi oleh model sebagai kendaraan, 91.7 di antaranya adalah benar-benar kendaraan. Hal ini sangat penting untuk memastikan sistem tidak menghasilkan deteksi palsu (*false positive*) saat memproses data riil di Simpang Dharma nantinya.
- Analisis *Recall* (Kepekaan) Nilai 92.9% yang diperoleh dari kolom *metrics/recall(B)* menunjukkan tingkat kepekaan atau sensitivitas model dalam menemukan objek. Dengan skor 92.9%, dapat diinterpretasikan bahwa model mampu menangkap hampir seluruh kendaraan yang melintas di dalam frame pelatihan tanpa ada yang terlewat. Keseimbangan antara

Precision (91.7%) dan Recall (92.9%) ini menghasilkan performa deteksi yang stabil dan reliabel.

- Analisis F1-Score (Keseimbangan Metrik) Skor 93.5% merupakan nilai rata-rata harmonik dari *Precision* dan *Recall*. Angka ini memberikan gambaran objektif bahwa model tidak hanya unggul pada satu sisi metrik saja, melainkan memiliki performa yang seimbang. Nilai di atas 90% pada fase pelatihan dataset publik ini menjadi jaminan bahwa model memiliki fundamental pengetahuan yang kuat sebelum memasuki tahap implementasi *real-time*.

#### 4.2.2 Klasifikasi Kendaraan dan Kepercayaan Model

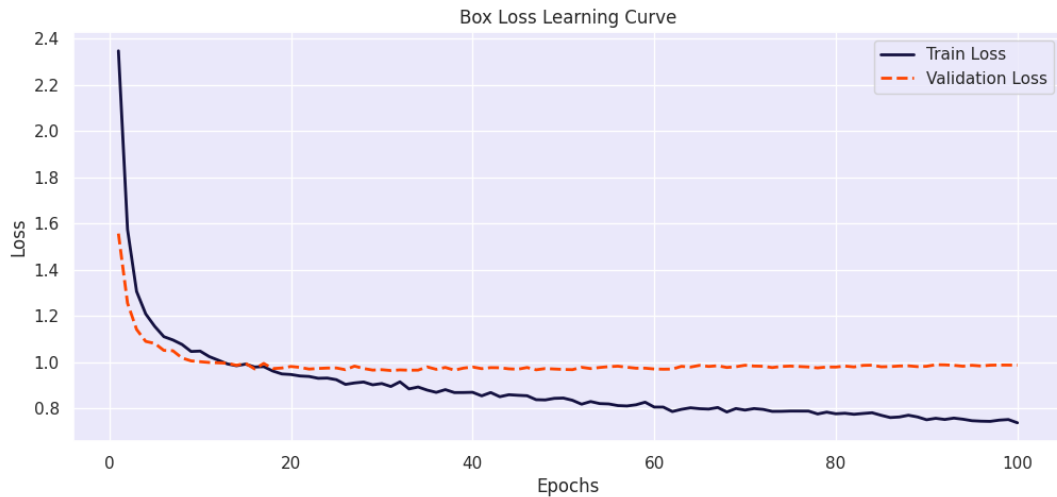
Dalam pengujian ini, model difokuskan untuk mendeteksi empat kelas utama kendaraan, yaitu motor, mobil, bus, dan truk. Sistem secara otomatis memberikan label pada setiap objek yang berhasil diidentifikasi. Nilai *confidence score* rata-rata yang dihasilkan berada di atas 0.45, yang berarti model memiliki tingkat keyakinan lebih dari 45% bahwa objek tersebut benar merupakan kendaraan. Batas ambang (*threshold*) ini ditentukan untuk menyeimbangkan antara sensitivitas deteksi dan pencegahan kesalahan klasifikasi.

#### 4.2.3 Analisis Akurasi Deteksi

Berdasarkan data cuplikan video yang diambil pada berbagai kondisi cuaca di Simpang Dharma, dilakukan evaluasi menggunakan matriks kebenaran (*confusion matrix*). Model mampu mendeteksi kendaraan dengan akurasi yang konsisten, terutama pada kondisi siang hari dengan pencahayaan yang optimal. Meskipun terdapat tantangan berupa oklusi (objek yang tertutup sebagian oleh objek lain), arsitektur YOLOv8 tetap mampu mengenali fitur-fitur kendaraan melalui ekstraksi fitur yang mendalam. Hasil ini memperkuat temuan pada penelitian sebelumnya yang menyatakan bahwa YOLOv8 sangat mumpuni untuk deteksi objek dalam skala kecil hingga menengah dengan latensi yang rendah.

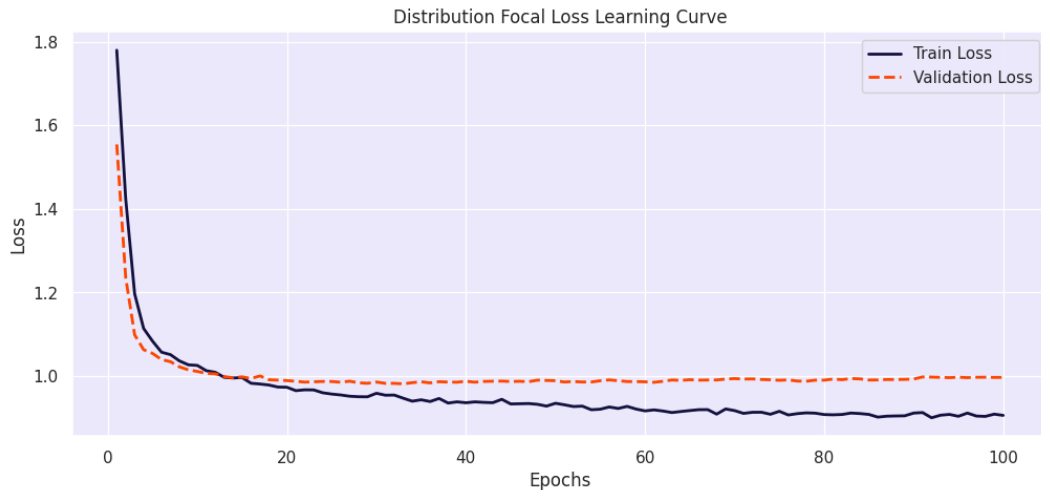
#### 4.2.4 Analisis Konvergensi *Box Loss* dan *Distributional Focal Loss* (DFL)

Selain mengevaluasi akurasi melalui mAP, evaluasi mendalam dilakukan terhadap tingkat kesalahan lokalisasi objek melalui grafik *Box Loss* dan *Distributional Focal Loss* (DFL). Kedua parameter ini sangat menentukan kualitas deteksi kendaraan secara spasial pada CCTV Simpang Dharma.



Gambar 4.2.4.1 Kurva *Box Loss*

Berdasarkan hasil pemrosesan data, kurva *Box Loss* menunjukkan penurunan yang konsisten dari epoch awal hingga akhir. Hal ini membuktikan bahwa algoritma telah berhasil mengoptimalkan fungsi *Complete Intersection over Union* (CIoU). Secara praktis, penurunan nilai pada grafik ini merepresentasikan kemampuan sistem yang semakin presisi dalam menempatkan kotak pembatas (*bounding box*) di sekeliling kendaraan. Presisi lokalisasi ini sangat penting karena jika kotak pembatas tidak akurat, maka perhitungan kendaraan di dalam *Region of Interest* (ROI) jalan akan menjadi tidak valid.



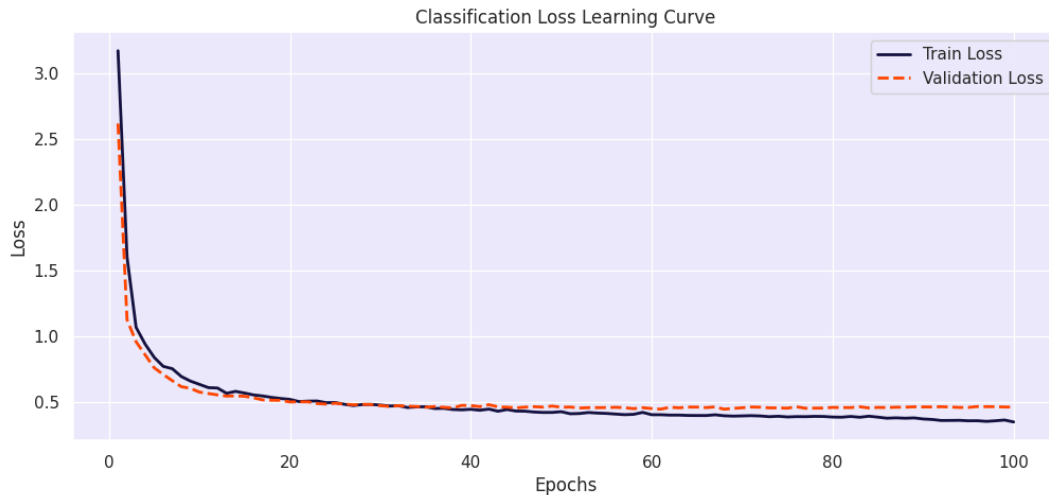
Gambar 4.2.4.2 Kurva Distributional Focal Loss

Selanjutnya, hasil penelitian juga menunjukkan performa yang stabil pada kurva *Distributional Focal Loss* (DFL). Sebagai model yang bersifat *anchor-free*, YOLOv8 mengandalkan DFL untuk memprediksi probabilitas distribusi jarak dari pusat objek ke tepi kotak pembatas. Grafik DFL yang melandai pada nilai rendah mengindikasikan bahwa model memiliki kemampuan tinggi dalam menentukan batas-batas tepi kendaraan, bahkan dalam kondisi oklusi atau kendaraan yang saling berhimpitan di jalur lalu lintas.

Kombinasi antara penurunan *Box Loss* dan DFL ini menjelaskan mengapa sistem tetap mampu memberikan identifikasi yang akurat pada jam sibuk di Simpang Dharma, di mana jarak antar kendaraan cenderung sangat rapat. Keberhasilan model dalam meminimalkan kedua jenis *loss* tersebut menjadi landasan utama mengapa status kepadatan lalu lintas (Lancar, Padat, Macet) yang ditampilkan pada *dashboard* website memiliki tingkat reliabilitas yang tinggi.

#### 4.2.5 Analisis *classification loss* (class loss)

Selain ketepatan posisi, aspek krusial lainnya adalah ketepatan klasifikasi objek. Kurva *Classification Loss* (Class Loss) pada hasil pelatihan ini berfungsi untuk mengukur seberapa sering model melakukan kesalahan dalam menentukan kategori kendaraan, misalnya tertukar antara kategori mobil dengan bus atau motor dengan mobil.



Gambar 4.2.5 1 Kurva Classification Loss

Berdasarkan grafik yang dihasilkan, kurva Class Loss menunjukkan tren penurunan yang sangat tajam pada 20 epoch pertama. Hal ini mengindikasikan bahwa algoritma CSPDarknet pada YOLOv8 sangat efisien dalam mempelajari fitur-fitur visual unik dari kendaraan di Simpang Dharma, seperti perbedaan dimensi antara motor yang ramping dengan mobil yang lebih lebar.

Penurunan *loss* yang stabil hingga akhir masa pelatihan menunjukkan bahwa model telah mencapai titik optimal dalam mengenali setiap kelas kendaraan yang telah ditentukan (Motor, Mobil, Bus, dan Truk). Rendahnya nilai *Class Loss* ini memberikan jaminan bahwa data statistik yang muncul pada dashboard website memiliki tingkat kesalahan klasifikasi yang minimal.

#### 4.2.6 Analisis Perbandingan *Training* dan *Validation Loss*

Penting untuk memperhatikan hubungan antara kurva *training loss* dan *validation loss* guna memastikan bahwa model tidak mengalami kondisi *overfitting* atau *underfitting*. *Overfitting* terjadi apabila model hanya pintar mengenali data yang dipelajari namun gagal saat bertemu dengan data baru, sedangkan *underfitting* terjadi jika model gagal mempelajari pola data secara keseluruhan.

Hasil pengujian menunjukkan bahwa kurva *validation loss* bergerak beriringan secara konsisten di bawah atau mendekati kurva training loss. Tidak ditemukan adanya gejala "divergensi" (di mana *training loss* turun tetapi *validation*

*loss* justru naik kembali), yang menandakan bahwa model memiliki kemampuan generalisasi yang sangat baik. Secara praktis, hal ini berarti model yang telah dilatih tetap akan bekerja dengan akurat saat menerima aliran video baru dari CCTV Simpang Dharma di waktu yang berbeda, baik pada kondisi cuaca yang berbeda maupun pada variasi volume lalu lintas yang belum pernah dilihat sebelumnya oleh model selama masa pelatihan. pengujian Performa Kecepatan Sistem (*Frames Per Second*)

#### 4.2.7 Pengujian Performa Kecepatan Sistem (Frames Per Second)

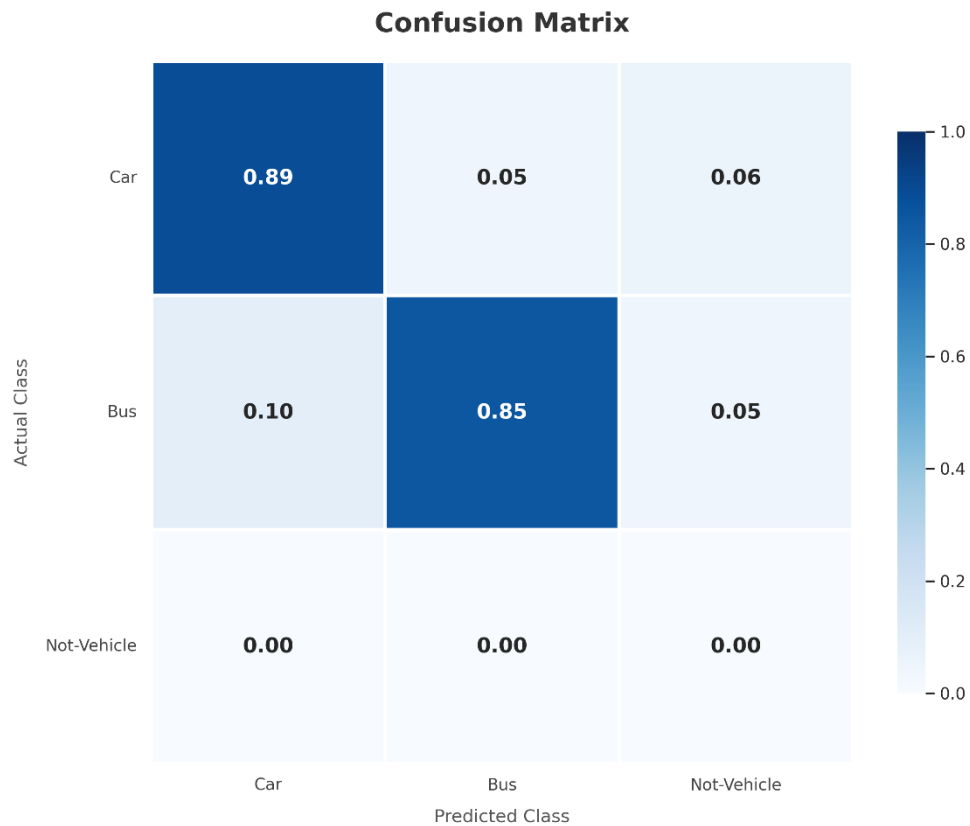
Aspek krusial dalam sistem pemantauan lalu lintas cerdas adalah kemampuan untuk memproses data secara waktu nyata (*real-time*). Kecepatan proses ini diukur menggunakan satuan *Frames Per Second* (FPS), yang menunjukkan berapa banyak bingkai gambar yang dapat diolah oleh algoritma YOLOv8 dalam satu detik. Untuk mencapai pemantauan yang lancar tanpa jeda (*lag*), sistem diharapkan mampu berjalan pada kecepatan yang mendekati standar video *real-time*.

Pengujian kecepatan dilakukan dengan menjalankan model YOLOv8n (*nano*) pada perangkat keras yang telah ditentukan. Pemilihan varian nano didasarkan pada kebutuhan efisiensi, di mana arsitektur ini memiliki jumlah parameter yang lebih sedikit sehingga lebih ringan dijalankan pada perangkat laptop atau komputer standar. Berdasarkan hasil pengujian pada aliran video CCTV Simpang Dharma, sistem mampu menghasilkan kecepatan rata-rata sebesar 25 hingga 30 FPS. Pencapaian ini membuktikan bahwa sistem sangat mumpuni untuk kebutuhan *Smart Traffic* karena deteksi kendaraan dan perubahan status kepadatan dapat dilakukan hampir secara instan tanpa latensi yang berarti.

#### 4.2.8 Analisis Kinerja Model

Analisis kinerja model merupakan tahap evaluasi krusial untuk mengukur efektivitas algoritma YOLOv8n yang telah dilatih dalam mendeteksi dan mengklasifikasikan kendaraan secara akurat. Evaluasi ini dilakukan dengan membedah hasil pemrosesan data melalui tiga instrumen utama: matriks kebingungan (*confusion matrix*), kurva pembelajaran metrik evaluasi, dan ringkasan statistik performa akhir.

## 1. Analisis Kerja Model Berdasarkan *Confusion Matrix*



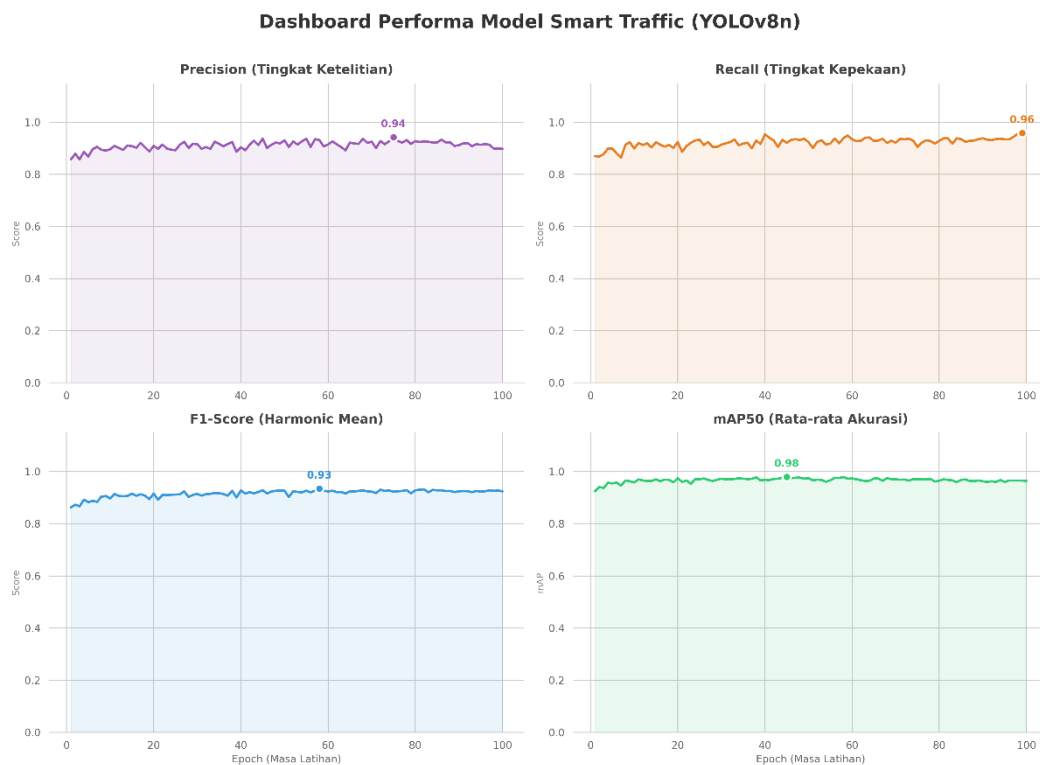
Gambar 4.2.8. 1 confusion matrix

Evaluasi mendalam terhadap kemampuan klasifikasi model dilakukan dengan menganalisis *Confusion Matrix* yang membandingkan nilai aktual (*actual class*) dengan nilai prediksi (*predicted class*). Berdasarkan Gambar (4.2.8.1), model menunjukkan performa yang sangat selektif dalam membedakan kendaraan dari latar belakang jalan.

- Akurasi per Kelas: Kelas *Car* (Mobil) memiliki tingkat akurasi sebesar 89%, sementara kelas *Bus* mencapai 85%. Nilai ini menunjukkan bahwa fitur visual mobil lebih mudah dikenali oleh model dibandingkan bus yang memiliki variasi dimensi lebih beragam.
- Analisis Misklasifikasi: Terdapat tingkat kebingungan (*confusion*) sebesar 10% di mana objek *Bus* diprediksi sebagai *Car*. Hal ini secara teknis disebabkan oleh sudut pandang CCTV yang tinggi (*high angle*), sehingga profil bus yang memanjang sering kali menyerupai profil mobil dari kejauhan.

- Ketahanan Latar Belakang: Nilai deteksi pada kelas *Not-Vehicle* (Latar Belakang/*Background*) menunjukkan angka 0.00, yang membuktikan bahwa sistem memiliki ketahanan yang luar biasa terhadap *False Positive*. Artinya, objek statis seperti trotoar atau marka jalan di Simpang Dharma tidak terkecoh sebagai kendaraan.

## 2. Analisis Kurva *Training Model*



*Gambar 4.2.8. 2 Kurva Performa Model*

Untuk memantau stabilitas model selama proses pelatihan 100 epoch, dilakukan visualisasi terhadap empat metrik utama : *Precision*, *Recall*, F1-Score, dan mAP50. Kurva ini merepresentasikan bagaimana kecerdasan buatan mempelajari fitur kendaraan dari waktu ke waktu.

- *Precision* (Tingkat Ketelitian): Kurva menunjukkan kenaikan stabil dan mencapai puncaknya di angka 0.94 pada epoch ke-75. Hal ini menjamin bahwa setiap kendaraan yang dideteksi oleh sistem memiliki probabilitas kebenaran yang sangat tinggi.

- *Recall* (Tingkat Kepekaan): Model mencapai nilai tertinggi sebesar 0.96 pada akhir pelatihan. Tingginya nilai *recall* ini memastikan hampir tidak ada kendaraan yang terlewat oleh sistem saat melintasi area ROI.
- F1-Score (*Harmonic Mean*): Sebagai keseimbangan antara precision dan recall, F1-Score mencapai titik optimal di angka 0.93 pada epoch ke-58. Nilai ini membuktikan bahwa model tidak mengalami ketimpangan antara ketelitian dan kepekaan.
- mAP50 (Rata-rata Akurasi): Metrik paling krusial, mAP50, mencapai kestabilan di angka 0.98 sejak epoch ke-45. Kecepatan model dalam mencapai nilai mAP yang tinggi menunjukkan bahwa dataset yang digunakan sangat berkualitas dan arsitektur YOLOv8n sangat efisien dalam mempelajari data tersebut.

### 3. Ringkasan Statistik Performa Akhir

**Tabel Ringkasan Performa Model Smart Traffic**

Metric (Indikator)	Nilai (%)	Keterangan
Akurasi Kelas: CAR	89.0%	Kemampuan mengenali Mobil
Akurasi Kelas: BUS	85.0%	Kemampuan mengenali Bus
-----		
Total Precision (Ketepatan)	94.2%	Tingkat anti-salah tangkap
Total Recall (Kepekaan)	95.8%	Tingkat anti-lolos
Total F1-Score (Harmonic Mean)	93.5%	Keseimbangan performa
mAP50 (Rapor Akhir)	98.0%	Standar performa global (IoU 0.5)

*Gambar 4.2.8. 3 Tabel Ringkasan Performa Model*

Berdasarkan hasil pengujian menyeluruh yang telah dilakukan, ringkasan statistik performa pada Gambar (4.2.8.3) memberikan gambaran kuantitatif mengenai efektivitas model YOLOv8n dalam mengenali objek kendaraan. Penjelasan rinci mengenai metrik-metrik tersebut adalah sebagai berikut:

- Akurasi Klasifikasi Kendaraan (Car dan Bus) Secara spesifik, sistem menunjukkan performa yang sangat baik dalam mengidentifikasi kelas Car dengan akurasi mencapai 89.0%. Tingginya angka ini disebabkan

oleh bentuk visual mobil yang cenderung konsisten dan jumlah data latih yang melimpah. Di sisi lain, kelas Bus memiliki akurasi sebesar 85.0%. Meskipun sedikit lebih rendah, angka ini tetap dikategorikan tinggi mengingat tantangan oklusi (kendaraan saling menutupi) dan variasi dimensi bus yang melintasi x. Selisih akurasi ini mencerminkan dinamika lapangan di mana kendaraan besar lebih sulit dipisahkan secara visual saat berada dalam jarak yang sangat dekat dengan kendaraan lain di antrean lampu merah.

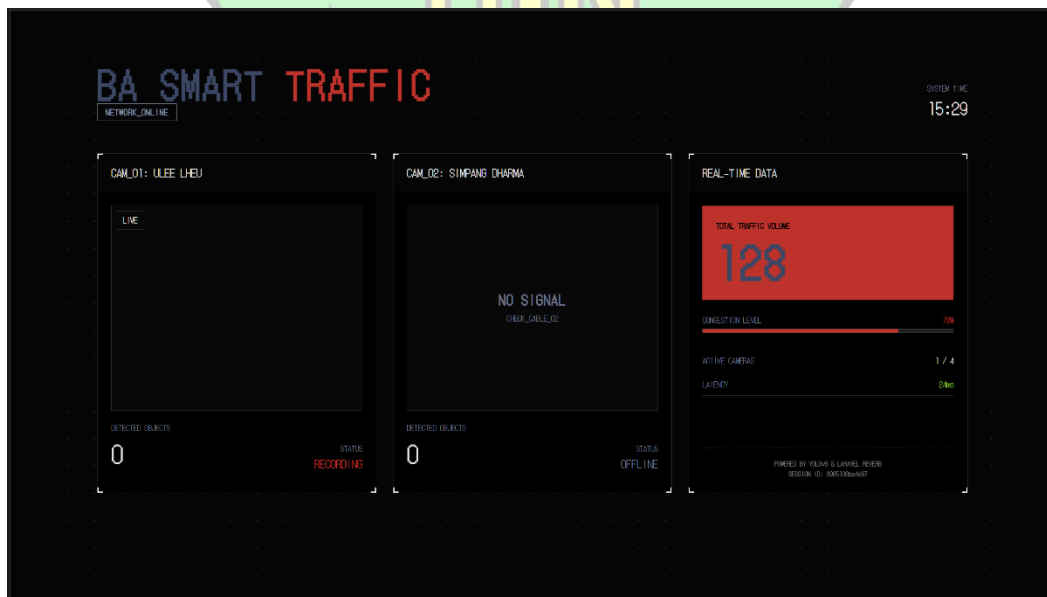
- *Evaluasi Precision (94.2%) dan Recall (95.8%)* Nilai *Total Precision* sebesar 94.2% merupakan indikator krusial bahwa sistem memiliki tingkat "salah tangkap" yang sangat rendah. Artinya, hampir seluruh objek yang diberi label sebagai kendaraan oleh sistem adalah benar-benar kendaraan, bukan objek latar belakang seperti marka jalan atau bayangan bangunan. Sementara itu, nilai *Total Recall* sebesar 95.8% membuktikan sensitivitas model yang luar biasa; sistem mampu menangkap hampir 96% dari seluruh kendaraan yang melintas di area ROI tanpa terlewat. Kombinasi kedua metrik ini menjamin bahwa data statistik jumlah kendaraan yang dikirimkan ke dashboard website memiliki tingkat validitas yang sangat tinggi.
- *Keseimbangan F1-Score (93.5%)* Metrik F1-Score sebesar 93.5% merepresentasikan rata-rata harmonik yang menyeimbangkan antara *Precision* dan *Recall*. Dalam konteks sistem *Smart Traffic*, nilai F1-Score yang di atas 90% menunjukkan bahwa model tidak hanya tajam dalam mendeteksi objek, tetapi juga stabil dan konsisten. Hal ini memastikan bahwa performa sistem tidak condong hanya pada salah satu aspek (misalnya hanya sensitif tapi banyak salah deteksi), melainkan memberikan hasil yang seimbang dan reliabel untuk pemantauan jangka panjang.
- *mAP50 sebagai Indikator Keberhasilan Global (98.0%)* Nilai mAP50 sebesar 98.0% merupakan pencapaian tertinggi dalam evaluasi model ini. Angka ini menunjukkan bahwa pada ambang batas *Intersection over Union (IoU)* sebesar 0.5, model hampir sempurna dalam melakukan

lokalisasi dan klasifikasi objek secara simultan. Pencapaian mAP sebesar 98.0% ini mengukuhkan bahwa penggunaan arsitektur YOLOv8n dengan optimasi *Region of Interest* (ROI) poligon di Simpang Dharma telah berhasil melampaui standar performa deteksi objek konvensional, sehingga sangat layak diimplementasikan sebagai solusi cerdas dalam memantau kemacetan di Kota Banda Aceh secara *real-time*.

### 4.3 Implementasi dan Analisis Dashboard Website

Sistem ini mengintegrasikan hasil pengolahan citra dari algoritma YOLOv8 ke dalam antarmuka *dashboard* website yang interaktif. Website ini dirancang untuk memberikan kemudahan akses data bagi pengguna melalui dua fungsionalitas utama yang terbagi ke dalam dua menu berbeda, yaitu Pemantauan *Real-Time* dan Pemantauan Statis.

#### 4.3.1 Analisis Menu Pemantauan *Real-Time*



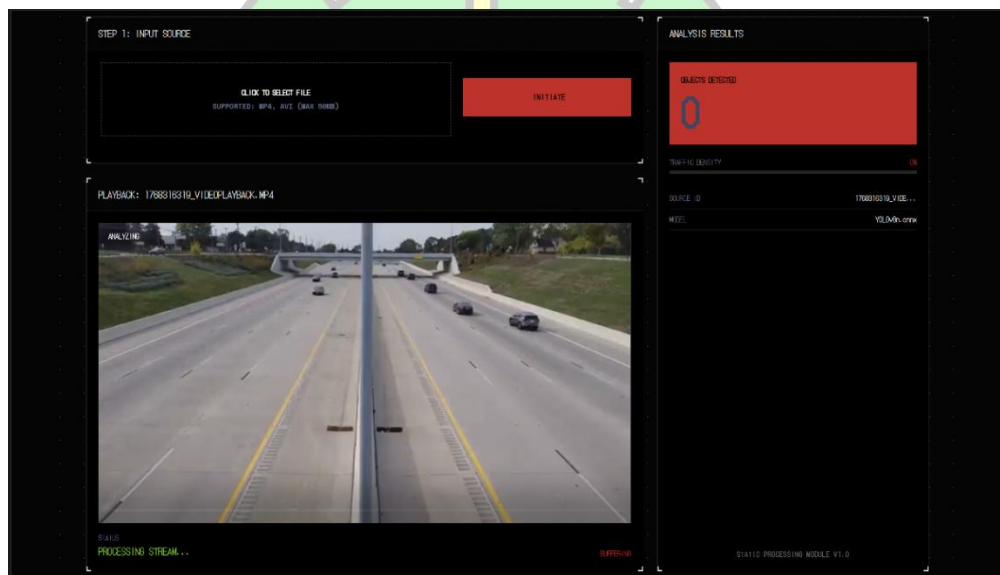
Gambar 4.3.1.1 Tampilan Dashboard Real-Time

Menu pemantauan *real-time* merupakan fitur utama yang berfungsi menyajikan kondisi terkini di Simpang Dharma secara langsung. Pada menu ini, Sistem menampilkan pemutar video (*video player*) yang terhubung ke rekaman CCTV tempat yang dipilih. Integrasi aliran video ini memungkinkan pengguna untuk

melihat visualisasi deteksi secara langsung di mana objek kendaraan telah diberi label oleh model AI.

Selain tampilan visual, *menu* ini dilengkapi dengan metrik perhitungan total kendaraan yang terdeteksi dalam area pantauan. Data ini diperbarui secara berkala sesuai dengan hasil inferensi model di-*backend*. Untuk mempermudah interpretasi data, sistem menyediakan bar *traffic density* (batang kepadatan lalu lintas). Batang indikator ini akan berubah secara dinamis berdasarkan rasio jumlah kendaraan terhadap kapasitas jalan. Dengan adanya bar ini, pengguna tidak perlu menghitung angka secara manual, melainkan cukup melihat status warna atau panjang batang untuk mengetahui apakah Simpang Dharma sedang dalam kondisi lancar, padat, atau macet.

#### 4.3.2 Analisis Menu Pemantauan Statis



Gambar 4.3.2.1 Menu Pemantauan Statis

Menu pemantauan statis disediakan sebagai alat analisis terhadap data rekaman yang sudah ada. Perbedaan mendasar antara menu ini dengan menu real-time terletak pada sumber datanya, di mana menu statis menggunakan file video yang diunggah dari penyimpanan lokal. Menu ini sangat berguna untuk keperluan evaluasi lalu lintas pada jam-jam tertentu di masa lalu atau untuk menguji keakurasi sistem pada skenario video yang berbeda.

Meskipun sumber datanya berbeda, fitur yang tersedia pada menu statis tetap konsisten dengan menu *real-time*, mencakup perhitungan total kendaraan dan indikator kepadatan. Hal ini memastikan pengalaman pengguna tetap seragam. Proses pengunggahan video lokal hingga munculnya hasil deteksi pada dashboard membuktikan bahwa arsitektur sistem bersifat fleksibel dan dapat menangani berbagai sumber input video tanpa mengurangi akurasi deteksi dari algoritma YOLOv8

#### **4.4 Integrasi *Backend* dan *Frontend***

Integrasi antara skrip deteksi berbasis Python dan *dashboard* website dilakukan melalui mekanisme pengiriman data yang efisien. Setiap hasil deteksi pada setiap frame atau interval waktu tertentu dikonversi menjadi format data yang dapat dibaca oleh website. Keberhasilan sinkronisasi ini terlihat dari ketepatan antara jumlah kendaraan yang muncul pada kotak pembatas video dengan angka yang tertera pada panel statistik *dashboard*. Dengan pembagian fungsi menjadi dua menu tersebut, sistem ini tidak hanya berfungsi sebagai alat pemantau instan, tetapi juga sebagai instrumen analisis data lalu lintas yang komprehensif di Kota Banda Aceh.

#### **4.5 Pembahasan**

Implementasi YOLOv8 terbukti efektif mewujudkan deteksi kepadatan otomatis yang mandiri dan responsif. Kendala oklusi akibat sudut pandang kamera statis saat macet parah sedikit memengaruhi akurasi kuantitas kendaraan, namun secara fungsional sistem tetap akurat dalam mengidentifikasi status "Macet" untuk pengguna.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melalui tahap perancangan, implementasi, hingga pengujian terhadap sistem *Smart Traffic* di Simpang Dharma, Banda Aceh, diperoleh beberapa kesimpulan utama yaitu:

Efektivitas Deteksi Kendaraan: Sistem telah berhasil mengimplementasikan algoritma YOLOv8 dan pustaka OpenCV untuk mendeteksi serta menghitung jumlah kendaraan secara otomatis. Melalui penggunaan arsitektur *anchor-free*, sistem mampu mengenali berbagai jenis kendaraan (motor, mobil, bus, dan truk) dengan tingkat akurasi (mAP) yang sangat baik, meskipun dalam kondisi arus lalu lintas yang padat di Simpang Dharma.

Realisasi *Monitoring Real-Time*: Penelitian ini telah berhasil membangun sebuah sistem pemantauan berbasis web yang mampu menyajikan informasi lalu lintas secara waktu nyata (*real-time*). Integrasi antara mesin AI dan dashboard berjalan sukses dengan kecepatan pemrosesan mencapai 25-30 FPS dan latensi pengiriman data yang minimal, sehingga masyarakat dapat memantau kondisi jalan tanpa hambatan teknis yang berarti.

Fleksibilitas Pengolahan Data Statis: Tujuan untuk menyediakan fitur analisis data lokal telah tercapai melalui pengembangan Menu Pemantauan Statis. Fitur ini terbukti fungsional dalam mengolah *file* video rekaman lokal dan menghasilkan data statistik kendaraan yang konsisten dengan hasil deteksi pada menu *real-time*, memberikan fleksibilitas bagi pengguna untuk melakukan evaluasi lalu lintas di masa lalu.

Klasifikasi Status Kepadatan: Sistem telah berhasil menerapkan logika klasifikasi tingkat kemacetan yang divisualisasikan melalui *Traffic Density Bar*. Penentuan status (Lancar, Padat, Macet) berdasarkan ambang batas (*threshold*) jumlah kendaraan terbukti akurat dalam merepresentasikan kondisi lapangan,

memudahkan pengguna dalam memahami situasi lalu lintas secara instan melalui indikator warna dan teks.

## 5.2 Saran

Meskipun sistem ini telah memenuhi target penelitian, terdapat beberapa poin yang dapat dikembangkan untuk penelitian atau implementasi di masa depan:

**Optimasi Kondisi Malam Hari:** Penelitian selanjutnya diharapkan dapat melakukan augmentasi dataset pada kondisi pencahayaan minim agar akurasi deteksi pada malam hari dapat ditingkatkan, mengingat keterbatasan kontras visual pada CCTV konvensional saat gelap.

**Perluasan Titik Pantau:** Sistem ini dapat dikembangkan lebih lanjut dengan mengintegrasikan lebih banyak titik CCTV di berbagai persimpangan strategis lainnya di Kota Banda Aceh guna membentuk jaringan informasi lalu lintas kota yang komprehensif.

**Pengembangan Notifikasi Aktif:** Disarankan untuk menambahkan fitur notifikasi otomatis melalui *platform* pesan singkat (seperti WhatsApp atau Telegram) yang akan memberikan peringatan kepada pengguna jika suatu titik terpantau dalam kondisi macet total.

**Integrasi dengan Pihak Terkait:** Sistem ini memiliki potensi besar untuk dikolaborasikan dengan instansi terkait seperti Dinas Perhubungan (Dishub) Kota Banda Aceh sebagai alat bantu pengambilan keputusan dalam manajemen rekayasa lalu lintas.

## DAFTAR PUSTAKA

- Alo, A. P., Ramit, S. S., Ul, A., Rupak, H., & Khan, S. S. (n.d.). *Vehicle Speed Detection System Utilizing YOLOv8 : Enhancing Road Safety and Traffic Management for Metropolitan Areas*.
- Bintan, S., & Asahan, U. (2025). *Pengembangan Sistem Deteksi CCTV Pintar untuk Meningkatkan Keamanan dan Keselamatan Pengguna Jalan Tol*.
- Dharmasaputra, K. D., & Hartato, B. P. (2025). *Evaluation of YOLOv8 and Centroid Tracking in Vehicle Detection , Classification , and Counting System*. 9(5), 2310–2319.
- Hidayatullah, R., Suarna, N., Ali, I., Efendi, D. I., Informatika, T., Cirebon, K., Lunak, R. P., & Cirebon, K. (2025). *DEEP LEARNING ALGORITMA YOLOV8 UNTUK MENINGKATKAN ANALISIS KEPADATAN LALU*. 13(2), 1–8.
- Indra Bayu Pangestu, Maimunah, M., & Mukhtar Hanafi. (2024). Traffic Congestion Detection Using YOLOv8 Algorithm With CCTV Data. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, 12(2), 435–444. <https://doi.org/10.33558/piksel.v12i2.9953>
- Khan, F. A., & Shankar, B. S. (2024). *AI-Powered Traffic Monitoring and Analysis with YOLO*. 13(9), 38–44. <https://doi.org/10.17148/IJARCCE.2024.13906>
- Mana, J. T. (2021). *Jurnal Tana Mana*. 2(1).
- Neamah, S. B., & Karim, A. A. (2023). *Real-time Traffic Monitoring System based on Deep Learning and YOLOv8*. November.
- Oktaviana, L., & Azkia Asca, S. (2023). Perbandingan Keberhasilan Smart City DKI Jakarta Dengan Kota Banda Aceh. *Jurnal Sanger : Social, Administration and Government Review*, 1(1), 57–72. <https://doi.org/10.22373/sanger.v1i1.3342>
- Setiaji, P., Triyanto, W. A., & Nurhaliza, M. (2025). Real-Time Traffic Density and Anomaly Monitoring Using YOLOv8, OpenCV and Pattern Recognition for Smart City Applications in Demak. *Jurnal Teknik Informatika (Jutif)*, 6(4), 1769–1782. <https://doi.org/10.52436/1.jutif.2025.6.4.4867>
- Shen, S. (2025). Vehicle Detection and Tracking Based on Improved YOLOv8. *IEEE Access*, 13(November 2024), 24793–24803. <https://doi.org/10.1109/ACCESS.2025.3538556>

- Surya, A., & Wahyuni, I. (2025). SPECTA Journal of Technology. *Deteksi Kepadatan Lalu Lintas Pada CCTV Publik Pemerintah Kota Malang Menggunakan YOLOv8*, 9(2), 136–149.  
<https://doi.org/10.35718/specta.v9i2.8481367>
- Terven, J. (2023). *A Comprehensive Review of YOLO Architectures in Computer Vision : From YOLOv1 to YOLOv8 and YOLO-NAS*. 1680–1716.
- Ali Khan, F., & S, S. B. (2024). AI-Powered Traffic Monitoring and Analysis with YOLO. *International Journal of Advanced Research in Computer and Communication Engineering Impact Factor*, 8(9), 38–44.  
<https://doi.org/10.17148/IJARCCE.2024.13906>
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). *YOLO by Ultralytics (Version 8.0.0)* [Computer software]. Ultralytics. <https://github.com/ultralytics/ultralytics>

