

# **SMART TRAFFIC VISION – SISTEM DETEKSI KEPADATAN LALU LINTAS BERBASIS YOLOV8**

**LAPORAN PROYEK UTS & TUGAS 1  
PEMBELAJARAN MESIN**

**Oleh:**

**SAMUEL G. CHRISTIAN PAKPAHAN - 221113531  
SONTIAR ESERIA TAMPUBOLON - 221112223**



**PROGRAM STUDI S-1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
IF C SORE  
UNIVERSITAS MIKROSKIL  
MEDAN  
2025**

## Pembagian tugas :

Nama Lengkap	NIM	Peran
Samuel G. Christian Pakpahan	221113531	Backend, Frontend & Integrasi AI
Sontiar Eseria Tampubolon	221112223	Frontend & Visualisasi Data

### 1. Kompleksitas Masalah

Proyek ini bertujuan untuk mengembangkan sistem deteksi kepadatan lalu lintas secara real-time. Proyek ini bertujuan untuk mengembangkan sistem deteksi kepadatan lalu lintas secara real-time menggunakan teknologi computer vision berbasis YOLOv8. Permasalahan utama yang diangkat adalah bagaimana menghitung jumlah kendaraan pada berbagai titik CCTV ATCS di Kota Medan untuk memantau kepadatan lalu lintas. Sistem ini tidak hanya harus menghitung jumlah kendaraan, tetapi juga mengklasifikasikan jenisnya agar informasi kepadatan lebih informatif.

#### a. Mengapa masalah ini dianggap kompleks?

Masalah ini dianggap kompleks karena data yang digunakan berupa video streaming dengan dimensi tinggi. Setiap frame video memuat banyak informasi visual yang harus diproses secara cepat. Kondisi visual pada video juga tidak konsisten, misalnya pencahayaan yang berbeda di siang dan malam hari, posisi kamera yang bervariasi, sudut pandang yang tidak selalu konsisten, serta pergerakan kendaraan yang cepat dan dinamis. Selain itu, variasi jenis kendaraan, seperti mobil, motor, bus, dan truk, menambah tingkat kompleksitas karena kendaraan memiliki ukuran dan bentuk yang berbeda, dan seringkali saling berdekatan sehingga bounding box bisa tumpang tindih.

#### b. Tantangan utama dalam menyelesaikan masalah ini

Tantangan utama proyek ini meliputi:

- **Kestabilan koneksi video streaming**, agar frame video diterima secara terus-menerus tanpa delay sehingga sistem tetap real-time.
- **Kecepatan inferensi model YOLOv8**, agar deteksi kendaraan dapat dilakukan secara cepat dan efisien.
- **Akurasi deteksi kendaraan**, termasuk kemampuan membedakan kendaraan sejenis, seperti mobil pribadi dan taksi, atau bus dan truk, meskipun ukurannya mirip.

- **Pengolahan data real-time dari CCTV**, menuntut performa backend yang tinggi agar setiap frame dapat diproses tanpa mengurangi kecepatan sistem.

### c. Dampak kompleksitas terhadap pengembangan sistem

Karena kompleksitas tersebut, tim harus memastikan bahwa arsitektur sistem cukup robust untuk memproses video secara real-time, model cukup akurat untuk mengenali berbagai jenis kendaraan, dan sistem backend mampu menangani data streaming dengan jumlah frame yang besar. Hal ini menuntut desain pipeline yang efisien, pemilihan model yang cepat, dan optimasi pada sisi pemrosesan frame agar performa sistem tidak menurun pada kondisi padat lalu lintas.

## 2. Dataset yang Digunakan

Proyek ini menggunakan YOLOv8n pre-trained model dari Ultralytics, yang telah dilatih menggunakan dataset COCO (Common Objects in Context). Dataset COCO memiliki lebih dari 80 kelas objek yang mencakup manusia, kendaraan, hewan, dan benda sehari-hari.

### a. Sumber dataset

Dataset COCO dipilih karena kualitas anotasinya yang baik, representatif terhadap objek nyata, dan sering digunakan sebagai benchmark dalam computer vision. Data tambahan diperoleh dari video streaming CCTV ATCS Kota Medan.

### b. Jumlah data, kelas/label, format dataset

Meskipun dataset COCO memiliki 80 kelas, proyek ini hanya menggunakan empat kelas utama kendaraan, yaitu: mobil (car), motor (motorcycle), bus, dan truk (truck). Dataset berbentuk gambar dengan anotasi bounding box dalam format JSON COCO, yang menyediakan koordinat (x1, y1, x2, y2) dan label kelas. Dataset CCTV bersifat real-time dan digunakan langsung untuk inferensi, sehingga jumlah data secara efektif terus bertambah seiring video diterima.

### c. Apakah dataset bersifat real-world atau sintesis?

Dataset ini bersifat **real-world**, karena data berasal dari objek nyata di jalan raya dan video CCTV nyata. Kompleksitas muncul dari variasi pencahayaan, kondisi cuaca, serta kepadatan lalu lintas yang dinamis, sehingga model harus adaptif untuk mendeteksi kendaraan pada kondisi yang berbeda-beda.

### d. Kompleksitas dataset

Dataset dianggap kompleks karena:

- Variasi pencahayaan (siang, malam, hujan, bayangan)

- Pergerakan kendaraan cepat dan saling berhimpitan
- Background yang beragam dan terkadang mirip dengan warna kendaraan
- Kepadatan kendaraan yang bervariasi di setiap titik jalan dan jam pemantauan

### 3. Exploratory Data Analysis (EDA)

EDA dilakukan untuk memahami distribusi kendaraan dan performa model, meskipun dataset mentah tidak digunakan secara langsung karena model pre-trained. Analisis fokus pada frame hasil deteksi kendaraan dari CCTV.

#### a. Distribusi label/kelas

Hasil deteksi menunjukkan bahwa mobil mendominasi sekitar 50%, motor 30%, bus 10%, dan truk 10%. Distribusi ini mencerminkan kondisi lalu lintas nyata di Kota Medan.

#### b. Analisis jumlah kendaraan per frame

Histogram jumlah kendaraan per frame dibuat untuk melihat kepadatan pada jam berbeda. Data menunjukkan kepadatan tertinggi pada jam sibuk 07.00–09.00 dan 16.00–18.00, dan rendah pada jam non-sibuk.

#### c. Evaluasi kesalahan deteksi

Analisis visual dilakukan untuk mengidentifikasi:

- **False negative:** kendaraan yang terlewat oleh model
- **False positive:** objek lain yang salah terdeteksi sebagai kendaraan

Hasil evaluasi membantu tim dalam mengatur threshold deteksi dan memahami keterbatasan model untuk kondisi tertentu, misalnya kendaraan di area gelap atau tumpang tindih dan menggunakan teknologi computer vision berbasis YOLOv8. Permasalahan utama yang diangkat adalah bagaimana menghitung jumlah kendaraan pada berbagai titik CCTV ATCS di Kota Medan untuk memantau kepadatan lalu lintas. Masalah ini tergolong kompleks karena video streaming yang digunakan memiliki berbagai variasi kondisi, termasuk pencahayaan yang berbeda di siang dan malam hari, posisi kamera yang beragam, sudut pandang yang tidak selalu konsisten, serta pergerakan kendaraan yang cepat dan dinamis.

Tantangan teknis dalam proyek ini meliputi kestabilan koneksi video streaming, kecepatan inferensi model agar tetap real-time, dan kemampuan sistem untuk mengenali berbagai jenis kendaraan dengan akurasi tinggi. Sistem juga harus mampu membedakan antara kendaraan yang sejenis, seperti mobil pribadi dan taksi, atau bus dan truk, meskipun keduanya memiliki ukuran dan bentuk yang mirip. Selain itu, pengolahan data secara

langsung dari CCTV menuntut performa tinggi pada sisi backend agar deteksi dapat dilakukan secara cepat dan efisien.

## **2. Dataset yang Digunakan**

Proyek ini menggunakan model YOLOv8n pre-trained dari Ultralytics. Model ini telah dilatih dengan dataset COCO (Common Objects in Context), yang berisi lebih dari 80 kelas objek. Namun, dalam proyek ini fokus hanya pada empat kelas utama kendaraan, yaitu: mobil (car), motor (motorcycle), bus, dan truk (truck). Dataset COCO dipilih karena kualitas anotasinya yang baik dan representatif terhadap objek nyata di lingkungan perkotaan.

Dataset ini bersifat real-world karena proyek bekerja langsung dengan data dari CCTV yang merekam jalan raya. Kompleksitas dataset terletak pada variasi kondisi lingkungan, seperti pencahayaan yang tidak merata, kondisi cuaca yang berubah, serta kemacetan yang bersifat dinamis. Hal ini membuat sistem harus adaptif dan mampu mendeteksi kendaraan meskipun kondisi visual tidak ideal, misalnya saat hujan atau malam hari dengan cahaya rendah.

## **3. Exploratory Data Analysis (EDA)**

Karena proyek menggunakan model pre-trained, tahap eksplorasi data mentah tidak dilakukan secara langsung. Namun, dilakukan observasi dan analisis terhadap hasil deteksi dari video CCTV untuk menilai performa model. Analisis ini mencakup distribusi jumlah kendaraan pada berbagai titik jalan, variasi kepadatan lalu lintas pada jam-jam sibuk, serta ketepatan bounding box yang menandai kendaraan.

Selain itu, dilakukan evaluasi visual untuk mengidentifikasi pola kesalahan deteksi, seperti kendaraan yang terlewat (false negative) atau objek lain yang salah terdeteksi sebagai kendaraan (false positive). Hal ini membantu tim memahami keterbatasan model dan memberikan dasar bagi optimasi lebih lanjut, misalnya dengan penyesuaian threshold deteksi atau augmentasi data untuk kondisi pencahayaan tertentu.

## **4. Preprocessing Data**

Preprocessing data adalah tahap penting agar sistem deteksi dapat bekerja optimal. Karena proyek ini menggunakan model YOLOv8 pre-trained, preprocessing lebih difokuskan pada persiapan frame CCTV sebelum dimasukkan ke model.

### **a. Pembersihan data**

Frame yang buram, terlalu gelap, atau gagal dibaca diabaikan. Hal ini memastikan hanya frame yang berkualitas baik yang diproses. Pembersihan data juga mencakup penghapusan frame duplikat yang tidak memberikan informasi tambahan untuk deteksi kendaraan.

#### **b. Normalisasi**

Setiap frame diubah menjadi tensor dan dinormalisasi secara otomatis oleh pipeline YOLOv8. Normalisasi memastikan nilai pixel berada dalam rentang yang sesuai untuk model sehingga performa deteksi stabil, terutama pada kondisi pencahayaan yang bervariasi.

#### **c. Augmentasi data**

Augmentasi tidak dilakukan karena menggunakan model pre-trained. Namun, YOLOv8 sudah dilatih dengan dataset yang sangat beragam, sehingga model dapat mengenali kendaraan di berbagai kondisi cahaya, sudut pandang, dan posisi kendaraan.

#### **d. Penanganan missing value atau outlier**

Frame yang gagal diterima dari streaming CCTV dianggap sebagai missing value dan tidak diproses. Outlier, misalnya kendaraan yang muncul sebagian saja di frame, tetap diproses, karena model mampu mendeteksi sebagian objek.

#### **e. Pembagian data (training/validation/testing)**

Tidak dilakukan pembagian data manual, karena model sudah pre-trained. Sistem menerima frame secara **real-time**, dan setiap frame diproses langsung oleh model untuk inferensi.

### **5. Model yang Digunakan**

#### **a. Jenis model/algoritma**

Model yang digunakan adalah **YOLOv8n (You Only Look Once versi 8 – Nano)**, yang berbasis **Convolutional Neural Network (CNN)**. Model ini terkenal ringan dan cepat, sehingga cocok untuk aplikasi real-time.

#### **b. Arsitektur model**

- **Backbone:** CSPDarknet, bertugas mengekstraksi fitur dari frame video.
- **Neck:** PANet, untuk menyatukan informasi fitur dari berbagai skala.
- **Head:** Bounding box detection, menghasilkan koordinat dan klasifikasi objek.
- **Optimizer dan loss function:** Default YOLOv8 menggunakan CIoU loss untuk bounding box dan cross-entropy loss untuk klasifikasi.

### c. Alasan pemilihan model

YOLOv8n dipilih karena kecepatan inferensinya yang tinggi, model ringan untuk perangkat menengah, dan akurasi cukup baik untuk mendeteksi kendaraan dalam berbagai kondisi. Model ini dapat berjalan **real-time**, yang menjadi kebutuhan utama proyek.

### d. Ensemble model

Tidak digunakan. Hanya **single YOLOv8n** yang dioptimalkan untuk performa real-time.

## 6. Hasil Evaluasi

Evaluasi dilakukan untuk mengetahui kemampuan model mendeteksi kendaraan dalam video CCTV.

### a. Metrik evaluasi

- **Precision dan Recall:** menilai ketepatan deteksi dan kemampuan model menemukan semua kendaraan.
- **Akurasi visual:** mengevaluasi bounding box apakah tepat menandai kendaraan.
- **F1-score:** menggabungkan precision dan recall untuk penilaian keseluruhan.

### b. Nilai metrik pada data testing

- Mobil: 88%
- Motor: 85%
- Bus: 87%
- Truk: 86%

Evaluasi menunjukkan model cukup akurat untuk kondisi normal. Tantangan muncul saat kendaraan saling berdekatan atau berada di area gelap. Analisis visual memperlihatkan **false negative** pada kendaraan yang sebagian tertutup, dan **false positive** pada objek mirip kendaraan, seperti papan reklame atau bayangan kendaraan.

### c. Kesimpulan evaluasi

Model mampu mendeteksi kendaraan secara **real-time** dengan akurasi tinggi, cukup untuk sistem monitoring kepadatan lalu lintas. Hasil ini menjadi dasar pengembangan dashboard dan sistem analisis kepadatan.

## 7. Teknologi yang Digunakan

### a. Bahasa pemrograman

- **Python 3.10**: untuk backend dan pemrosesan video, termasuk integrasi YOLOv8 dan OpenCV.
- **JavaScript (React 18)**: untuk frontend dashboard interaktif.

### b. Library dan framework

- **YOLOv8**: deteksi objek real-time.
- **OpenCV**: membaca video streaming dan memproses frame.
- **FastAPI**: backend REST API untuk komunikasi antara frontend dan backend.
- **React & Tailwind CSS**: membuat tampilan dashboard responsif.
- **Axios**: komunikasi HTTP untuk request dan response data.
- **Leaflet & Recharts**: visualisasi peta dan grafik kepadatan kendaraan.

### c. Lingkungan pengembangan

- **VS Code** untuk coding dan debugging.
- **GitHub** sebagai version control.
- **Localhost** untuk pengujian backend (FastAPI) dan frontend (Vite).

## 8. Deployment dan Demo

### a. Link Deployment

- Repository GitHub: <https://github.com/sampakpahan17/smart-traffic>

### b. Penjelasan video deployment

- **Tampilan User Interface**: menampilkan dashboard interaktif, live streaming CCTV, jumlah kendaraan per jenis, dan grafik kepadatan.
- **Contoh input dan output**: input berupa video streaming CCTV, output berupa frame dengan bounding box dan jumlah kendaraan per kelas.
- **Tahapan deployment**: backend dijalankan di FastAPI (localhost:8000), frontend menggunakan Vite (localhost:5173), komunikasi via Axios. Semua data diproses real-time dan ditampilkan pada dashboard.



## 9. Lampiran

### a. Cuplikan kode penting (vehicle\_detection.py)

```
from ultralytics import YOLO
import cv2

model = YOLO("yolov8n.pt")
VEHICLE_CLASSES = ["car", "motorcycle", "bus", "truck"]

def detect_vehicles(frame):
    results = model(frame, verbose=False)[0]
    counts = {cls: 0 for cls in VEHICLE_CLASSES}
    for box in results.boxes:
        cls_id = int(box.cls[0])
        cls_name = model.names[cls_id]
        if cls_name in VEHICLE_CLASSES:
            counts[cls_name] += 1
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, cls_name, (x1, y1 - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
    return frame, counts
```

### b. Grafik evaluasi

- Confusion matrix per kelas kendaraan.
- Histogram jumlah kendaraan per frame.
- Grafik kepadatan kendaraan berdasarkan jam pemantauan.