

**DETEKSI HELM DAN PLAT SEPEDA MOTOR MENGGUNAKAN YOLOV8  
UNTUK PENCATATAN PELANGGAR LALU LINTAS**

**SKRIPSI**

**YUSRIANTONI**

**171402145**



**PROGRAM STUDI S1-TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

DETEKSI HELM DAN PLAT SEPEDA MOTOR MENGGUNAKAN YOLOV8 UNTUK  
PENCATATAN PELANGGAR LALU LINTAS

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Teknologi Informasi

YUSRIANTONI

171402145



PROGRAM STUDI S1-TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024

## PERSETUJUAN

Judul : DETEKSI HELM DAN PLAT SEPEDA MOTOR  
MENGUNAKAN YOLOV8 UNTUK PENCATATAN  
PELANGGAR LALU LINTAS

Kategori : Skripsi

Nama : Yusriantoni

Nomor Induk Mahasiswa : 171402145


Program Studi : Teknologi Informasi

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA


Medan, 12 Juli 2024

Komisi Pembimbing:

Pembimbing 2

  
Mohammad Fadly Syah Putra M.Sc  
198301292009121003

Pembimbing 1

  
Dedy Arisandi S.T., M. Kom  
199003272017062001

Diketahui/disetujui oleh  
Teknologi Informasi  
Ketua,

  
Dedy Arisandi S.T., M. Kom  
199003272017062001



## **PERNYATAAN**

### **DETEKSI HELM DAN PLAT SEPEDA MOTOR MENGGUNAKAN YOLOV8 UNTUK PENCATATAN PELANGGAR LALU LINTAS**

#### **SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali ada beberapa kutipan dan ringkasan yang masing – masing telah disebutkan sumbernya.

Medan,

Yusriantoni

171402145

## UCAPAN TERIMA KASIH

Terima kasih kepada:

1. Keluarga penulis, terutama kedua orang tua penulis Ibu Inn Kesuma dan Bapak Chandra Yu yang telah membesarkan saya, serta memberikan kasih sayang, doa, semangat, dan bimbingan hingga penulis bisa menyelesaikan tugas akhir ini.
2. Saudara-saudara penulis yaitu Yushelia, Yudhi Johansen, dan Yushellyn Valerina yang telah memberikan semangat, doa, dan dukungan.
3. Bapak Dr. Muryanto Amin, S.Sos, M.Si. yang merupakan Rektor Universitas Sumatera Utara.
4. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Dedy Arisandi S.T., M.Kom. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
6. Bapak Dedy Arisandi, ST., M.Kom. selaku Dosen Pembimbing 1 dan Bapak Mohammad Fadly Syah Putra M.Sc. selaku Dosen Pembimbing 2 yang telah memberikan bimbingan, saran dan motivasi dan waktu yang diluangkan untuk membantu menyelesaikan skripsi ini.
7. Dosen pembeding
8. Seluruh dosen Program Studi S1 Teknologi Informasi yang telah memberikan ilmu bagi penulis.
9. Pegawai dan staff Fakultas Ilmu Komputer dan Teknologi Informasi USU yang mendukung segala urusan administrasi dan akademik selama perkuliahan.
10. Teman-teman TI yang telah membantu penulis dalam menyelesaikan masa perkuliahan.

## ABSTRAK

Meningkatnya mobilitas masyarakat menuntut adanya solusi efektif untuk meningkatkan keselamatan di jalan raya, khususnya bagi pengendara sepeda motor. Tingginya tingkat pelanggaran lalu lintas dan kecelakaan yang sering berakibat fatal menjadi perhatian serius. Dalam konteks ini, teknologi computer vision, khususnya object detection, menawarkan potensi besar untuk menghadapi tantangan keselamatan ini. Penelitian ini bertujuan untuk mengembangkan sistem deteksi helm dan plat sepeda motor menggunakan metode YOLOv8 (You Only Look Once version 8), yang dikenal dengan kecepatan dan akurasi yang baik. Implementasi sistem dilakukan menggunakan bahasa pemrograman Python dengan framework Flask. Hasil uji coba menunjukkan bahwa sistem yang dibangun mampu mendeteksi helm dengan akurasi 80,28%, pengendara sepeda motor dengan akurasi 90,26%, dan plat sepeda motor dengan akurasi 76,8%. Namun, akurasi ekstraksi plat sepeda motor masih rendah, yaitu 27,5%. Meskipun demikian, sistem ini telah berfungsi dengan baik dalam mendeteksi helm, pengendara sepeda motor, dan plat sepeda motor, namun masih diperlukan peningkatan lebih lanjut terutama dalam ekstraksi plat sepeda motor. Penelitian ini diharapkan dapat berkontribusi dalam menciptakan lingkungan lalu lintas yang lebih aman dan tertib.

Kata kunci: helm, pengendara sepeda motor, plat sepeda motor, YOLOv8

## **HELMET AND MOTORCYCLE PLATE DETECTION USING YOLOV8 FOR TRAFFIC VIOLATION RECORDING**

### **ABSTRACT**

The increasing of mobility of the society demanding an effective approach to improve the road safety, mainly for motorcyclists. The high rate of traffic violations and frequent fatal accident have become a serious concern. In this case, computer vision technology, specifically object detection, offers a significant potential to address these safety challenges. This study aims to develop a helmet and motorcycle license plate detection system using YOLOv8 method, it is known for its good speed and accuracy. The system was implemented using Python programming language and Flask framework. The test results shows that the system can detect helmets with an accuracy of 80,28%, motorcyclists with an accuracy of 90,26%, and motorcycle license plates with an accuracy of 76,8%. However, the accuracy of license plate extraction is still very low, at 27,5%. Despite this, the system has performed well in detecting helmets, motorcyclists, and motorcycle license plates, but there is still a way to improve the system especially for the license plate extraction. This research is expected to contribute to creating a safer and more organized traffic environment.

**Keywords:** helmet, motorcyclist, license plate, YOLOv8

## Daftar Isi

PERSETUJUAN	iii
PERNYATAAN	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vi
ABSTRACT	vii
Daftar Isi	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB 1	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Tujuan Penelitian	4
1.4. Batasan atau Ruang Lingkup Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.6.1. Studi Literatur	4
1.6.2. Analisis Permasalahan	5
1.6.3. Perancangan Sistem	5
1.6.4. Implementasi	5
1.6.5. Pengujian	5
1.6.6. Penyusunan Laporan	5
1.7. Sistematika Penulisan	5
BAB 2	7
LANDASAN TEORI	7
2.1. Helm	7
2.1.1. Helm Full Face	7
2.1.2. Helm <i>Half Face</i>	7
2.1.3. Helm <i>Open Half</i>	8
2.2. Computer Vision	8
2.3. Open CV	9



2.4. Object Detection	9
2.5. YOLO (You Only Look Once)	10
2.5.1. YOLOv8	11
2.6. Matrix	14
2.6.1. IoU	14
2.6.2. Confusion Matrix	15
2.7. Penelitian Terdahulu	16
2.7.1. Perbedaan dengan penelitian terdahulu	19
<b>BAB 3</b>	<b>21</b>
<b>ANALISIS DAN PERANCANGAN SISTEM</b>	<b>21</b>
3.1. Analisis Sistem	21
3.2. Image Acquisition	22
3.3. Preprocessing	23
3.3.1. Resizing	23
3.3.2. Labeling	23
3.4. Model You Only Look Once v8 (YOLOv8)	24
3.4.1. Clone repository	24
3.4.2. Install requirements	24
3.4.3. Menambahkan data.yaml	24
3.4.4. Proses training	24
3.4.4. Proses testing	27
3.5. Perancangan Antarmuka Sistem	30
3.5.1. Rancangan Tampilan Utama	30
3.5.2. Rancangan Tampilan Live Stream	31
3.5.3. Rancangan Tampilan upload video	32
<b>BAB 4</b>	<b>33</b>
<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>33</b>
4.1. Implementasi Sistem	33
4.1.1. Perangkat keras dan lunak yang digunakan	33

4.1.2. Implementasi Rancangan Antar Muka	34
4.1.3. Implementasi Data	35
4.2. Training model YOLOv8	37
4.2.1. Hasil training model deteksi helm	37
4.2.2. Hasil Training model deteksi pengendara sepeda motor	39
4.2.3. Hasil training model deteksi plat sepeda motor	40
4.3. Prosedur Operasional	41
4.4. Pengujian Sistem	42
BAB 5	55
KESIMPULAN DAN SARAN	55
5.1. Kesimpulan	55
5.2. Saran	55
DAFTAR PUSTAKA	56

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Helm Full Face .....	7
<b>Gambar 2. 2</b> Helm Half Face.....	8
<b>Gambar 2. 3</b> Helm Open Half.....	8
<b>Gambar 2. 4</b> Arsitektur YOLOv8.....	12
<b>Gambar 2. 5</b> Visualisasi anchor box YOLOv8.....	13
<b>Gambar 2. 7</b> Head YOLOv8.....	13
<b>Gambar 2. 6</b> Head YOLOv8.....	14
<b>Gambar 2. 8</b> IoU (Padilla et al., 2021) .....	15
<b>Gambar 3. 1</b> Arsitektur Umum.....	22
<b>Gambar 3. 2</b> Cara Labeling .....	24
<b>Gambar 3. 3</b> Rancangan Tampilan Utama .....	31
<b>Gambar 3. 4</b> Rancangan Tampilan Live Stream .....	32
<b>Gambar 3. 5</b> Rancangan Tampilan Upload File .....	32
<b>Gambar 4. 1</b> Tampilan Utama .....	34
<b>Gambar 4. 2</b> Tampilan Live Stream .....	35
<b>Gambar 4. 3</b> Tampilan Upload File.....	35
<b>Gambar 4. 4</b> Data training dan Validation Helmet.....	36
<b>Gambar 4. 5</b> Data training dan Validation Driver .....	37
<b>Gambar 4. 6</b> Hasil training model deteksi helm .....	38
<b>Gambar 4. 7</b> Hasil training model deteksi pengendara sepeda motor .....	39
<b>Gambar 4. 8</b> Hasil training model deteksi plat sepeda motor.....	41
<b>Gambar 4. 9</b> Cara Menjalankan Aplikasi .....	41
<b>Gambar 4. 10</b> Alamat Lokal Flask .....	42

## DAFTAR TABEL

<b>Tabel 2. 1</b> Tabel Penelitian Terdahulu .....	19
<b>Tabel 4. 1</b> Tabel Akurasi Proses Training.....	42
<b>Tabel 4. 2</b> Hasil Pengujian .....	52
<b>Tabel 4. 3</b> Nilai True Positive, False Positive, dan False Negative .....	53

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Meningkatnya mobilitas masyarakat telah memperjelas urgensi penanganan masalah keselamatan di jalan raya, terutama yang berkaitan dengan pengendara sepeda motor. Tingkat pelanggaran lalu lintas yang tinggi dan kejadian kecelakaan yang sering kali berakhir dengan akibat fatal menjadi perhatian serius yang memerlukan respons yang efektif. Kondisi ini tidak hanya mengancam nyawa pengguna jalan, tetapi juga menciptakan dampak besar pada stabilitas dan keberlanjutan sistem transportasi. Dalam konteks perkembangan zaman dan pesatnya kemajuan teknologi, terbuka peluang besar untuk menghadapi tantangan keselamatan ini dengan *computer vision*.

*Computer vision* adalah bidang penelitian yang bertujuan untuk memungkinkan komputer untuk memiliki kemampuan penglihatan yang mirip dengan manusia. Pada ranah yang lebih umum, tujuan utama dari *computer vision* adalah menggunakan informasi visual yang diterima untuk mendapatkan pemahaman yang lebih dalam tentang lingkungan sekitar.

Salah satu pengaplikasian *computer vision* adalah *object detection*. *Object detection* adalah sebuah teknologi yang digunakan komputer untuk melihat dan mengenali apa yang ada di dalam sebuah gambar atau video. Teknologi ini tidak hanya menemukan objek, tetapi juga memberi tahu jenisnya (misalnya, orang, mobil, meja) dan lokasinya di dalam gambar.

Dengan memanfaatkan object detection untuk mengintegrasikan sistem deteksi helm, maka dapat mendeteksi apakah pengendara sepeda motor mematuhi peraturan penggunaan helm, dapat meningkatkan kepatuhan terhadap aturan keselamatan yang ada, dan dengan adanya deteksi nomor plat kendaraan dapat mengidentifikasi kendaraan yang terlibat dalam pelanggaran, membantu menciptakan lingkungan lalu lintas yang lebih aman dan tertib. Penerapan teknologi ini diharapkan mampu membawa perubahan positif dalam perilaku pengendara, memberikan insentif untuk kepatuhan, dan pada akhirnya, mengurangi angka kecelakaan di jalan raya.

Dengan memanfaatkan potensi teknologi, kita dapat merancang langkah-langkah proaktif yang tidak hanya mengatasi masalah keselamatan saat ini tetapi juga menciptakan fondasi yang lebih kokoh untuk masa depan mobilitas yang aman dan efisien.

Adapun penelitian terdahulu yang membuat sistem untuk mendeteksi apakah pengendara motor memakai helm atau tidak yang dilakukan oleh Albert et al (2020) dimana penelitian tersebut menggunakan metode CNN (*Convolutional Neural Network*) dan akurasi yang didapat adalah sebesar 70,49%.

Penelitian lainnya yang dilakukan Widiawati, C. R. A. (2019) membuat sebuah sistem untuk deteksi helm menggunakan metode *Automatic RoI dan Active Contour*. Dalam penelitian ini digunakan dataset yang berasal dari CCTV dengan jumlah 1000 dataset dan akurasi yang dihasilkan sebesar 72.97%.

Penelitian yang dilakukan oleh (Jamtsho et al., 2021) dimana peneliti membuat sistem deteksi plat sepeda motor dari pengendara motor yang tidak memakai helm secara real-time dengan metode YOLO dan menghasilkan akurasi sebesar 98.52%.

Penelitian lain dilakukan oleh Darmawan, H. (2022) membuat aplikasi untuk mendeteksi helm berbasis website dengan metode YOLOR (You Only Learn One Representaion). Dalam penelitian ini digunakan 1600 data yang kemudian diaugmentasi menjadi 2000 data dengan hasil akurasi sebesar 93%.

Kemudian penelitian oleh Mailoa et al (2022). Penelitian ini menggunakan metode You Only Look Once (YOLO) untuk mendeteksi bagian tubuh kepala dan

badan dari gambar yang diinput. Bagian tubuh yang terdeteksi dipotong dan diproses dengan metode *Convolutional Neural Network* (CNN) dan akurasi yang didapatkan adalah sebesar 96%.

Dalam penelitian ini, peneliti akan menggunakan YOLOv8. YOLOv8 merupakan salah satu model deteksi objek yang paling populer saat ini. YOLOv8 memiliki kecepatan dan akurasi yang bagus. Berikut adalah keunggulan menggunakan metode YOLOv8 (Yanto et al., 2023) :

1. Performa yang lebih tinggi

YOLOv8 menawarkan performa yang lebih unggul dibandingkan dengan versi sebelumnya dengan kemampuan mendeteksi objek yang lebih cepat dan akurat

2. Ukuran model yang lebih kecil

YOLOv8 memiliki ukuran model yang lebih kecil dibandingkan dengan versi sebelumnya sehingga lebih mudah diimplementasikan.

3. Peningkatan ekstraksi fitur:

Metode ekstraksi fitur pada YOLOv8 telah ditingkatkan, menghasilkan deteksi objek yang lebih akurat.

4. Kemampuan multiscale

YOLOv8 mampu menangani objek dengan berbagai ukuran dalam satu gambar yang sama.

5. Deteksi lebih banyak objek

YOLOv8 dapat mendeteksi banyak objek dalam satu gambar yang sama, sehingga dapat mempermudah proses deteksi.

6. Kemampuan deteksi pada gambar berukuran besar

YOLOv8 dapat mendeteksi objek pada gambar yang berukuran besar dengan akurasi yang tinggi.

Berdasarkan latar belakang dan penelitian sebelumnya, maka peneliti mengajukan sebuah penelitian untuk mendeteksi helm dan plat motor dengan metode

YOLOv8 untuk pencatatan pelanggar lalu lintas. Judul dari penelitian ini adalah "DETEKSI HELM DAN PLAT SEPEDA MOTOR MENGGUNAKAN YOLOV8 UNTUK PENCATATAN PELANGGAR LALU LINTAS".

## **1.2. Rumusan Masalah**

Pelanggaran terhadap aturan lalu lintas, seperti tidak mematuhi kewajiban memakai helm, masih sering terjadi di jalan raya dan berpotensi menimbulkan risiko dan bahaya yang dapat merugikan pengendara dan sekitarnya. Oleh karena itu dibutuhkan sistem untuk mendeteksi helm dan nomor plat sepeda motor agar bisa mencatat data pelanggar lalu lintas tersebut.

## **1.3. Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk mengembangkan sistem deteksi helm dan pencatatan plat nomor pelanggar lalu lintas yang lebih akurat dan efisien untuk meningkatkan kepatuhan terhadap peraturan lalu lintas terutama dalam hal penggunaan helm.

## **1.4. Batasan atau Ruang Lingkup Penelitian**

Batasan masalah dalam penelitian ini antara lain:

1. Helm yang dideteksi adalah helm motor seperti helm full face dan helm half face.
2. Citra diperoleh dari CCTV jalan raya.
3. Output yang dihasilkan adalah pengendara dengan helm, pengendara tanpa helm, dan nomor plat kendaraan
4. Karakter yang dideteksi hanya angka (0 – 9) dan huruf kapital (A – Z).

## **1.5. Manfaat Penelitian**

Dengan adanya penelitian ini diharapkan dapat :

- Meningkatkan kesadaran pengendara motor tentang pentingnya memakai helm di jalan raya.
- Mengurangi angka pelanggar lalu lintas terutama dalam hal penggunaan helm.

## **1.6. Metodologi Penelitian**

### **1.6.1. Studi Literatur**



Dalam tahap ini peneliti mempelajari dan mengumpulkan informasi dari berbagai sumber seperti jurnal buku dan artikel yang berhubungan dengan topik penelitian. Informasi tersebut adalah yang berkaitan dengan deteksi helm, deteksi plat motor, dan algoritma deteksi objek.

#### 1.6.2. Analisis Permasalahan

Dalam tahap ini dilakukan analisis permasalahan berdasarkan informasi yang telah diperoleh dari studi literatur, dan metode yang akan digunakan dalam penelitian ini adalah YOLOv8.

#### 1.6.3. Perancangan Sistem

Dalam tahap ini dilakukan pengumpulan dataset, membuat arsitektur umum, menentukan dataset yang akan digunakan, dan merancang sistem berdasarkan studi literatur dan analisis permasalahan.

#### 1.6.4. Implementasi

Penerapan dari perancangan sistem yang telah dibuat. Dalam penelitian ini, digunakan bahasa pemrograman python dan visual studio code sebagai text editor.

#### 1.6.5. Pengujian

Setelah tahap implementasi, akan dilakukan pengujian pada sistem yang telah dibuat untuk memastikan sistem tersebut sudah layak dipakai.

#### 1.6.6. Penyusunan Laporan

Peneliti akan menulis semua langkah-langkat dari penelitian yang telah dilakukan hingga selesai. Laporan ini dibuat berdasarkan data dan informasi yang diperoleh dari penelitian yang telah dilakukan.

### 1.7. Sistematika Penulisan

Berikut adalah sistematika penulisan penelitian yang terdiri dari 5 bab, yang masing-masing dijelaskan secara singkat dan disertai beberapa informasi berikut:

#### **BAB 1: Pendahuluan**

Bab ini memuat latar belakang, rumusan masalah, tujuan penelitian, Batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

**BAB 2: Landasan Teori**

Pada bab ini, penulis mengkaji teori-teori yang berkaitan dengan helm sepeda motor, computer vision, image processing, CUDA, Open CV, YOLO dan disertai penjelasan tentang metode YOLOv8.

**BAB 3: Analisis dan Perancangan Sistem**

Bab ini menjelaskan tahapan untuk membuat aplikasi deteksi helm dan plat sepeda motor sesuai dengan arsitektur umum yang telah dibuat.

**BAB 4: Implementasi**

Bab ini berisikan hasil implementasi dari model yang telah didapatkan melalui analisis dan desain sistem yang sudah dibangun.

**BAB 5: Kesimpulan dan Saran**

Bab ini berisi kesimpulan yang telah diperoleh dari penelitian ini dan saran untuk penelitian selanjutnya.

## BAB 2

### LANDASAN TEORI

#### 2.1. Helm

Helm adalah alat pelindung kepala yang biasanya dibuat dari metal atau bahan keras lainnya seperti kevlar, serat resin, atau plastik (Qoyyimah, 2023). Terdapat berbagai jenis helm sepeda motor, seperti helm *full-face*, *helm open half*, *helm half face*, *helm cross* dan berbagai jenis helm lainnya.

##### 2.1.1. Helm Full Face

Helm *full face* adalah jenis helm yang menutupi seluruh bagian wajah hingga dagu dengan kaca transparan di bagian depan untuk melindungi mata. Helm ini umumnya dipakai dalam balapan. Contoh helm *full-face* dapat dilihat pada Gambar 2.1.



**Gambar 2. 1** Helm *Full Face*

##### 2.1.2. Helm Half Face

Helm *half face* menyerupai helm *full face*, yang membedakannya adalah di bagian dagu helm *half face* tidak tertutup. Contoh helm *half face* dapat dilihat pada Gambar 2.2



**Gambar 2. 2** Helm *Half Face*

### 2.1.3. Helm *Open Half*

Helm *open half* memiliki bentuk yang menyerupai batok kelapa dan tidak dilengkapi dengan visor, sehingga mirip dengan helm sepeda biasa. Helm *open half* umumnya terbuat dari bahan plastic. Contoh helm *open half* dapat dilihat pada Gambar 2.3.



**Gambar 2. 3** Helm *Open Half*

## 2.2. Computer Vision

*Computer vision* adalah bidang penelitian yang bertujuan untuk memungkinkan komputer untuk memiliki kemampuan penglihatan yang mirip dengan manusia. Pada ranah yang lebih umum, tujuan utama dari *computer vision* adalah menggunakan informasi visual yang diterima untuk mendapatkan pemahaman yang lebih dalam tentang lingkungan sekitar. *Computer vision* memungkinkan computer untuk memiliki kemampuan untuk melihat seperti manusia.

Menurut Goodfellow et al. (2016) Computer Vision adalah bidang dalam kecerdasan buatan (AI) yang bertujuan untuk meniru kemampuan penglihatan manusia menggunakan kamera, komputer, dan algoritma. Teknologi ini memungkinkan

komputer untuk menafsirkan dan memahami dunia visual. Dengan menggunakan teknik pengolahan citra digital, computer vision dapat mengidentifikasi, melacak, dan memproses objek dalam gambar atau video. Implementasinya meliputi berbagai aplikasi seperti pengenalan wajah, pengenalan objek, dan analisis video.

*Computer vision* awalnya dibuat untuk memungkinkan komputer menginterpretasikan gambar dan video dalam konteks yang mirip dengan manusia. Pada intinya, computer vision memanfaatkan berbagai metode *machine learning* dan *deep learning* untuk menganalisis citra. Metode-metode ini melibatkan jaringan saraf tiruan (neural networks), terutama convolutional neural networks (CNN), yang mampu mengenali pola dalam data visual. CNN telah terbukti sangat efektif dalam tugas-tugas pengenalan citra, di mana mereka dapat dilatih untuk mengklasifikasikan berbagai objek dengan tingkat akurasi yang tinggi (Szeliski, 2010).

### **2.3. Open CV**

OpenCV (Open Source Computer Vision Library) adalah library computer vision dan machine learning yang bersifat open source. OpenCV dirancang untuk menyediakan infrastruktur umum bagi aplikasi visi komputer dan mempercepat penerapan persepsi mesin dalam produk komersial. Karena berlisensi BSD, OpenCV memungkinkan bisnis untuk menggunakan dan memodifikasi kodenya dengan mudah. Perpustakaan ini mencakup lebih dari 2500 algoritma yang dapat dioptimalkan, termasuk berbagai algoritma computer vision klasik serta algoritma computer vision dan machine learning yang terkini. OpenCV merupakan salah satu metode tercepat dan memiliki perpustakaan terlengkap untuk computer vision (Mistry & Saluja, 2016).

### **2.4. Object Detection**

*Object detection* merupakan sebuah teknologi yang digunakan komputer untuk melihat dan mengenali apa yang ada di dalam sebuah gambar atau video. Teknologi ini tidak hanya menemukan objek, tetapi juga memberi tahu jenisnya (misalnya, orang, mobil, meja) dan lokasi dari objek tersebut di dalam gambar.

Object detection adalah tugas mengidentifikasi dan menemukan objek dalam gambar atau video. Pendekatan yang lebih menantang diperlukan ketika berhadapan dengan gambar tiga dimensi yang kuat, seperti yang muncul dalam visi robot. Dalam keadaan ini, segmentasi sulit karena bayangan, nilai properti dan relasi ambigu jika orientasi objek tidak diketahui sebelumnya, dan bahkan jika objek tidak saling menutupi, hanya satu sisi objek yang dapat dilihat pada satu waktu. Melakukan segmentasi yang

memadai dalam keadaan ini mungkin memerlukan pembuatan peta kedalaman dari permukaan yang terlihat dalam adegan (menggunakan penginderaan rentang langsung, stereo, gerak, atau isyarat monokular), dan melakukan pengenalan objek mungkin memerlukan analisis kendala yang dikenakan gambar yang diamati pada kemungkinan objek dan kemungkinan orientasinya. (Rosenfeld, 1988)

## 2.5. YOLO (You Only Look Once)

YOLO adalah algoritma real-time object detection yang canggih. YOLO menerapkan convolutional neural network (CNN) ke seluruh gambar yang selanjutnya membagi gambar menjadi kotak-kotak kecil. Prediksi kotak pembatas dan confidence score masing-masing dihitung untuk setiap kotak. Bounding box ini dianalisis dengan confidence score yang diprediksi (Shinde et al., 2018).

YOLO memiliki beberapa keunggulan dibandingkan metode deteksi objek lainnya. Pertama, alih-alih menggunakan metode dua langkah untuk klasifikasi dan lokalisasi objek, YOLO menerapkan satu CNN untuk klasifikasi dan lokalisasi objek. Kedua, YOLO dapat memproses gambar dengan kecepatan sekitar 40-90 FPS, sehingga cukup cepat. Ini berarti streaming video dapat diproses secara real-time, dengan latensi yang dapat diabaikan dalam beberapa mili detik. Arsitektur YOLO membuatnya sangat cepat. Dibandingkan dengan R-CNN, YOLO 1000 kali lebih cepat dan 100 kali lebih cepat dari Fast R-CNN. (Redmon et al., 2017)

YOLO telah mengalami peningkatan sejak pertama kali diperkenalkan, diantaranya

YOLOv2, perbedaan YOLOv2 dengan versi sebelumnya adalah pada versi ini ditambahkan *anchor box* untuk meningkatkan akurasi deteksi (Gao et al., 2019). Arsitektur yang digunakan YOLOv2 lebih kompleks dari YOLOv1 dan memiliki lebih banyak layer. Oleh karena itu, YOLOv2 dapat mendeteksi objek dengan lebih cepat dan lebih akurat (Gao et al., 2019).

YOLOv3, pada versi ini pemrosesan multi skala diperkenalkan dan meningkatkan kinerja secara menyeluruh. YOLOv3 juga menjadi lebih akurat dalam mendeteksi objek terutama objek-objek kecil yang sulit dideteksi, bisa mendeteksi objek dengan resolusi gambar yang lebih tinggi, dan memakai anchor box yang lebih banyak untuk meningkatkan akurasi deteksi. YOLOv3 juga sudah bisa mendeteksi beberapa objek yang memiliki rasio ukuran yang berbeda pada sebuah gambar (Yanto et al, 2023).

YOLOv4, pada versi ini akurasi untuk mendeteksi objek ditingkatkan lagi. Perbedaan dengan versi YOLO sebelumnya adalah YOLOv4 menggunakan spatial pyramid pooling dan agregasi jalur PAnet pada bagian neck (Nepal & Eslamiat, 2022).

YOLOv5, (Yanto et al, 2023) menyatakan bahwa YOLOv5 hanya memperkenalkan model arsitektur yang baru dan dibuat dengan Bahasa pemrograman Python dan bisa diintegrasikann ke dalam sebuah sistem.

YOLOv6, dibandingkan dengan YOLOv5, YOLOv6 menggunakan arsitektur yang lebih baru dan memiliki lebih banyak layer. YOLOv6 juga menggunakan anchor box yang lebih rumit dibanding YOLOv5 untuk meningkatkan akurasi deteksi objek. Performa YOLOv6 dalam mendeteksi objek lebih baik akan tetapi membutuhkan waktu dalam proses komputasi (Yanto et al., 2023).

YOLOv7, berbeda dengan versi sebelumnya YOLOv7 memakai metode anchor-free dalam proses deteksi objek.

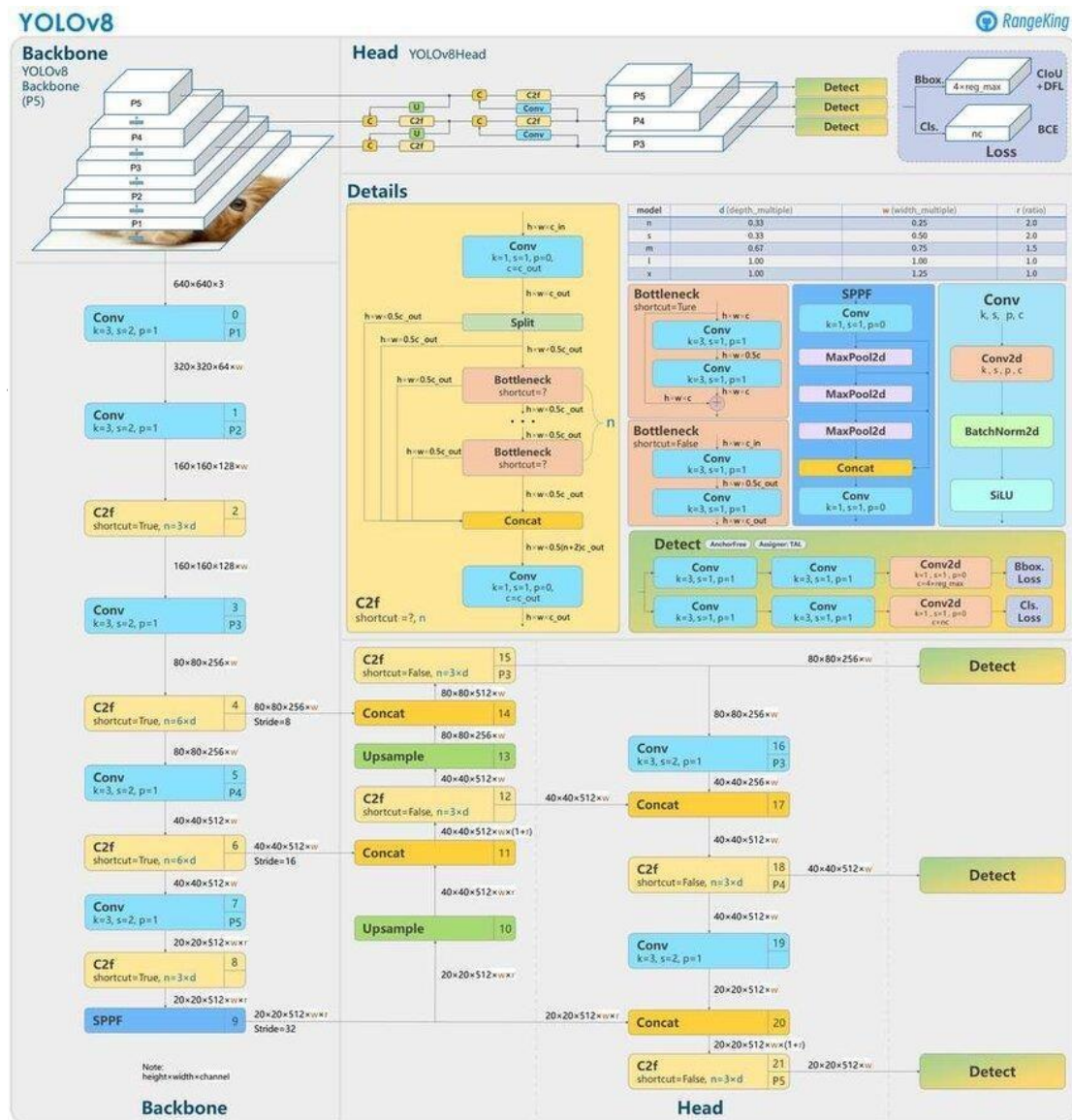
#### 2.5.1. YOLOv8

YOLOv8 adalah model deep-learning yang lebih meningkatkan kemampuan YOLOv5 di bidang object detection. Model ini menggabungkan berbagai struktur jaringan dan memanfaatkan modul C2F (Coarse-to-Fine) untuk meningkatkan YOLOv5. (Talib et al., 2024)

YOLOv8 menunjukkan akurasi yang luar biasa, melampaui versi YOLO sebelumnya. Algoritma YOLO dirancang khusus untuk mendeteksi dan mengklasifikasikan objek yang ada di suatu gambar. Namun, kinerjanya dalam mendeteksi objek – objek berukuran kecil mungkin relatif lebih rendah daripada algoritma tertentu ketika dikonfigurasi untuk beroperasi di lingkungan unik dengan dimensi tertentu. (Talib et al., 2024). Arsitektur YOLOv8 dapat dilihat di Gambar 2.4.

YOLOv8 dirilis pada tahun 2023. Versi ini membawa beberapa perbaikan dibandingkan pendahulunya, termasuk jaringan backbone yang lebih efisien, prediksi multiscale, dan sistem anchor yang baru. Struktur YOLOv8 terdiri dari tiga komponen utama: jaringan tulang punggung (backbone), leher (neck), dan kepala (head). Jaringan tulang punggung memanfaatkan Feature Pyramid Network (FPN) untuk mengekstraksi fitur dari gambar input. Bagian leher menggunakan serangkaian Cross-Layer Connection (CLC) untuk

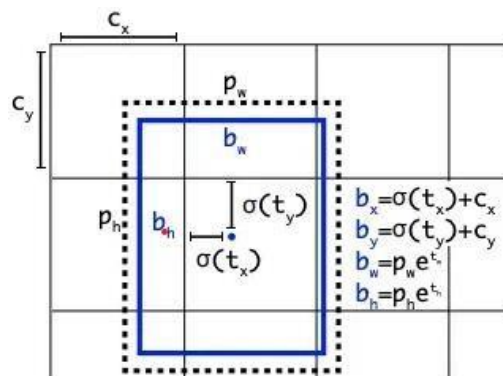
menyempurnakan fitur-fitur ini. Sementara itu, bagian kepala bertugas mengambil fitur yang telah disempurnakan untuk memprediksi bounding box, skor kelas objek, dan akurasi setiap objek dalam citra. YOLOv8 memiliki total 105 lapisan dalam arsitekturnya. Arsitektur YOLOv8 dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Arsitektur YOLOv8

Model YOLOv8 adalah anchor-free model yang berarti YOLOv8 memprediksi langsung pusat dari objek bukan offset dari anchor box.

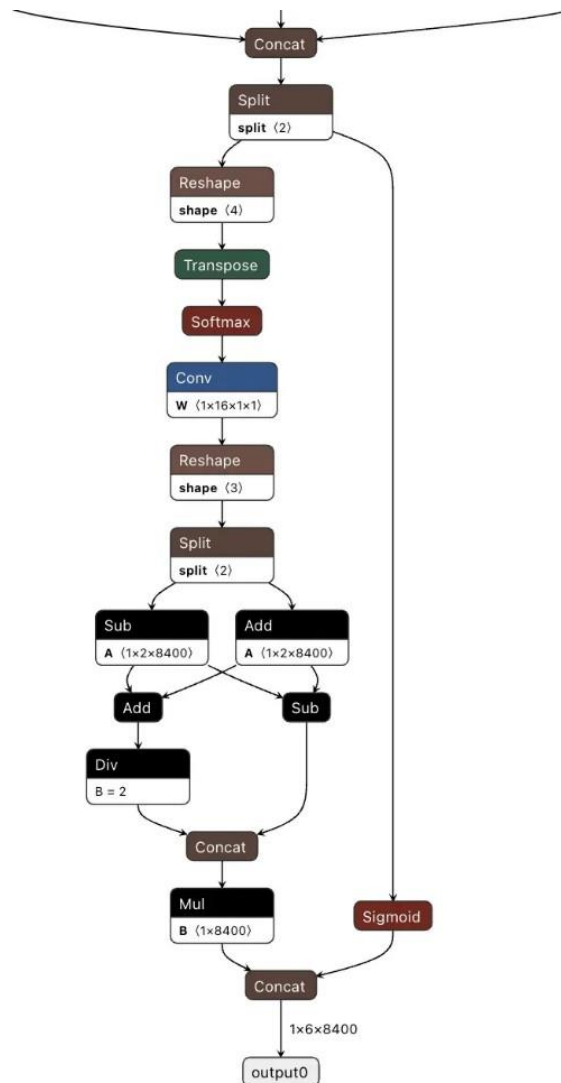




**Gambar 2. 5** Visualisasi anchor box YOLOv8

Anchor box pada model YOLO sebelumnya terkenal sulit untuk dikelola, karena meskipun mencerminkan distribusi kotak dari benchmark target, mereka sering kali tidak sesuai dengan distribusi dataset kustom.

Dengan tidak bergantung pada anchor box untuk mendeteksi, dapat mempercepat Non-Maximum Suppresion (NMS). NMS adalah Langkah post processing yang menyaring objek yang akan dideteksi.



**Gambar 2. 7** Head YOLOv8

## 2.6. Matrix

Dalam proses pengevaluasian sebuah model, dibutuhkan beberapa tolak ukur untuk menguji model tersebut. Diantaranya IoU dan Confusion Matrix.

### 2.6.1. IoU

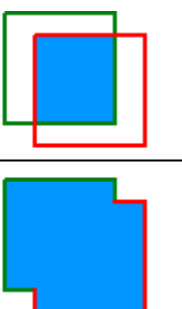
Intersection over Union (IoU) adalah metrik yang digunakan untuk menilai akurasi suatu model deteksi objek. Untuk menerapkan IoU dalam evaluasi model deteksi objek, diperlukan ground-truth bounding boxes dan predicted bounding boxes dari algoritma tersebut. Dengan memiliki kedua komponen ini, IoU dapat digunakan untuk

mengevaluasi kinerja aplikasi object detection. Untuk menghitung nilai IoU dapat dilakukan dengan cara

$$IoU = \frac{\text{area}(Bp \cap Bgt)}{\text{area}(Bp \cup Bgt)}$$

Dimana nilai Bp adalah area bounding box hasil prediksi dan Bgt adalah bounding box yang berasal dari ground truth box. Jika IoU bernilai 1, maka prediksi tersebut dikatakan sempurna dan jika bernilai 0, maka tidak terbentuk bounding box dari objek yang dideteksi.

Penggambaran IoU dapat dilihat pada gambar

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


**Gambar 2. 8** IoU (Padilla et al., n.d.)

### 2.6.2. Confusion Matrix

Confusion matrix adalah matrix yang menunjukkan jumlah data yang diprediksi, baik benar maupun salah. Dengan confusion matrix bisa didapatkan nilai recall, precision, dan akurasi. Confusion matrix terdiri atas 4 kategori yaitu true positive, true negative, false positive, dan false negative.

1. True positive (TP) adalah ketika model memprediksi positif dan yang sebenarnya juga positif
2. True negative (TN) adalah ketika model memprediksi negative dan yang sebenarnya juga negative
3. False positive (FP) adalah ketika model memprediksi positif dan yang sebenarnya negative

4. False negative (FN) adalah ketika model memprediksi negative dan data yang sebenarnya positif.

Dengan empat kategori tersebut, maka performa suatu model dapat dihitung dengan metrik evaluasi berikut:

1. Accuracy

Menghitung berapa banyak prediksi yang benar dari semua prediksi yang dilakukan oleh sebuah model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{total jumlah data}}$$

2. Recall

Mengukur kemampuan suatu model untuk menemukan semua contoh positif yang sebenarnya ada di dalam data.

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

3. Precision

Mengukur seberapa sering prediksi positif model benar dari semua prediksi positif

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

4. F1 Score

Menggabungkan precision dan recall menjadi satu metrik dengan mengambil rata-rata dari keduanya

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

## 2.7. Penelitian Terdahulu

Sejumlah penelitian terdahulu tentang deteksi helm dan deteksi plat sepeda motor sudah banyak dilakukan diantaranya penelitian oleh (Raras & Widiawati, 2019) menggunakan

1000 data yang didapat dari CCTV yang terdiri atas berbagai jenis kendaraan dimana peneliti membangun sebuah sistem untuk mendeteksi penggunaan helm pengendara sepeda motor menggunakan metode Automatic RoI dan Active Contour. Penelitian ini mendapat akurasi sebesar 72,97%.

Penelitian oleh (Pan et al., 2022) membuat sistem untuk mendeteksi plat sepeda motor di China secara real time menggunakan metode YOLOv7 dan LPRNet. Pada penelitian ini digunakan 1469 data dari CCPD sebagai data untuk diuji dan menghasilkan akurasi sebesar 96,1%.

Penelitian yang dilakukan oleh (Kulkarni et al., 2023) membangun sistem untuk mendeteksi helm, nomor plat kendaraan, dan kendaraan yang dicuri menggunakan YOLOv3, dan YOLOv5 dan menggunakan easyocr untuk ekstraksi nomor plat kendaraan. Data pelanggaran lalu lintas didapat dari rekaman CCTV setelah itu akan dicari frame dimana pelanggaran terjadi, dan diperbesar hingga terlihat nomor plat kendaraan. Untuk kendaraan yang dicuri, peneliti akan membandingkan nomor plat kendaraan tersebut dengan database. Jika sesuai, maka akan dikirim lokasi ke admin terdekat. Jika tidak sesuai maka kendaraan tersebut tidak dicuri.

(Gunadi & Setyati, 2020) menggunakan YOLOv3 untuk mendeteksi pengendara sepeda motor yang datanya diambil dari <https://pjreddie.com/> dan menggunakan Convolutional Neural Network (CNN) untuk mengklasifikasi pengguna helm. Akurasi yang didapat dari penelitian ini adalah 70,49%.

Penelitian yang dilakukan oleh (Jamtsho et al., 2021) dimana peneliti membuat sistem deteksi plat sepeda motor dari pengendara motor yang tidak memakai helm secara real-time dengan metode YOLO dan menghasilkan akurasi sebesar 98.52%.

Penelitian lain dilakukan oleh Darmawan H (2022) membuat aplikasi untuk mendeteksi helm berbasis website dengan metode YOLOR (You Only Learn One Representaion). Dalam penelitian ini digunakan 1600 data yang kemudian diaugmentasi menjadi 2000 data dengan hasil akurasi sebesar 93%.

Kemudian penelitian oleh Mailoa et al (2022). Penelitian ini menggunakan metode You Only Look Once (YOLO) untuk mendeteksi bagian tubuh kepala dan badan dari gambar yang diinput. Bagian tubuh yang terdeteksi dipotong dan diproses

dengan metode *Convolutional Neural Network* (CNN) dan akurasi yang didapatkan adalah sebesar 96%.

No	Penulis	Judul Penelitian	Tahun	Keterangan
1	Albert, Kartika Gunadi, Endang Setyati	Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network	2020	Peneliti melakukan deteksi sepeda motor dan pengendara sepeda motor menggunakan YOLO dan mengklasifikasi pengguna helm dan pengendara sepeda motor menggunakan CNN yang menghasilkan akurasi 70,49%
2	Rescky Marthen Mailoa, Leo Willyanto Santoso	Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN	2022	Peneliti menggunakan metode YOLO untuk menentukan posisi badan dan kepala dari gambar yang sudah diinput dan selanjutnya diklasifikasi dengan CNN dengan akurasi yang diperoleh untuk deteksi kepala adalah 64,09% dan badan 63,03%
3	Shenghu Pan, Jian Liu and Dekun Chen	Research on License Plate Detection and Recognition System based on YOLOv7 and LPRNet	2022	Peneliti membuat sistem untuk mendeteksi plat sepeda motor secara real time menggunakan YOLOv7 dan LPRNet. Penelitian ini menggunakan 1469 data dan mendapat hasil akurasi sebesar 96,%.

4	Chyntia Raras Ajeng Widiawati	Automatic RoI dan Active Contour untuk deteksi Penggunaan Helm pada Pengendara Sepeda Motor	2019	Peneliti melakukan deteksi helm dengan metode Automatic RoI dan Active Contour pada tahap segmentasi dan diklasifikasikan menggunakan Multilayer Perceptron classifier dan menghasilkan akurasi sebesar 72,97%
5	Lin, Hanhe Deng, Jeremiah D. Albers, Deike Siebert, Felix Wilhelm	Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi- Task Learning	2020	Penelitian ini menggunakan CNN-Based Multi-Task Learning terhadap 10006 data pengendara sepeda motor dari 12 wilayah di Myanmar dengan AP 95,3%

**Tabel 2. 1** Tabel Penelitian Terdahulu

#### 2.7.1. Perbedaan dengan penelitian terdahulu

Perbedaan dengan penelitian yang dilakukan oleh (Raras & Widiawati, 2019) adalah yaitu terdapat pada onjek yang dideteksi, dimana pada penelitian terdahulu objek yang dideteksi adalah berbagai jenis kendaraan seperti mobil dan sepeda motor, sedangkan pada penelitian ini hanya dilakukan deteksi helm dan plat sepeda motor. Perbedaan lainnya adalah perbedaan metode, metode yang digunakan dalam penelitian sebelumnya adalah Automatic RoI dan Active Contour dan metode yang digunakan dalam penelitian ini adalah YOLOv8.

Adapun perbedaan dengan penelitian yang dilakukan oleh (Pan et al., 2022) adalah terdapat pada sumber data, dan metode yang digunakan, Dimana pada penelitian sebelumnya digunakan dari data CCPD sebanyak 1469 data dan pada penelitian ini digunakan data yang direkam menggunakan kamera smartphone. Metode yang digunakan pada penelitian ini adalah YOLOv8 sedangkan pada penelitian sebelumnya menggunakan YOLOv7 dan LPRNET.

Kemudian perbedaan dengan penelitian yang dilakukan oleh (Kulkarni et al., 2023) adalah terdapat pada fungsi, metode dan sumber data yang digunakan. Pada penelitian sebelumnya digunakan metode YOLOv3, dan YOLOv5 untuk mendeteksi helm dan nomor plat kendaraan. Sedangkan pada penelitian ini digunakan metode YOLOv8 untuk mendeteksi helm dan plat sepeda motor pengendara yang tidak memakai helm. Untuk sumber data yang digunakan di penelitian sebelumnya digunakan data dari CCTV sedangkan untuk penelitian ini digunakan data yang direkam menggunakan kamera smartphone.

Selanjutnya perbedaan dengan penelitian yang dilakukan oleh (Gunadi & Setyati, 2020) adalah metode yang digunakan, objek yang dideteksi. Metode yang digunakan pada penelitian sebelumnya adalah YOLOv3 untuk mendeteksi pengendara sepeda motor, dan Convolutional Neural Network (CNN) untuk mengklasifikasi pengguna helm. Sedangkan pada penelitian ini digunakan metode YOLOv8 untuk mendeteksi helm dan pengendara sepeda motor yang tidak memakai helm.

Perbedaan penelitian dengan yang dilakukan oleh (Jamtsho et al., 2021) adalah penelitian sebelumnya terdapat pada metode yang digunakan dimana digunakan metode YOLO untuk penelitian sebelumnya. Sedangkan pada penelitian ini menggunakan YOLOv8 untuk mendeteksi helm dan pengendara sepeda motor yang tidak memakai helm.

Selanjutnya perbedaan penelitian ini dengan penelitian (Lin et al., 2020) terletak pada data dan metode yang digunakan. Pada penelitian ini digunakan data pengendara sepeda motor yang direkam di jalan raya dengan jumlah 2750 data dan metode yang digunakan pada penelitian ini adalah YOLOv8. Sedangkan pada penelitian sebelumnya dipakai data sebanyak 10006 data dari 12 wilayah yang berbeda di Myanmar dan menggunakan metode CNN-Based Multi-Task Learning.



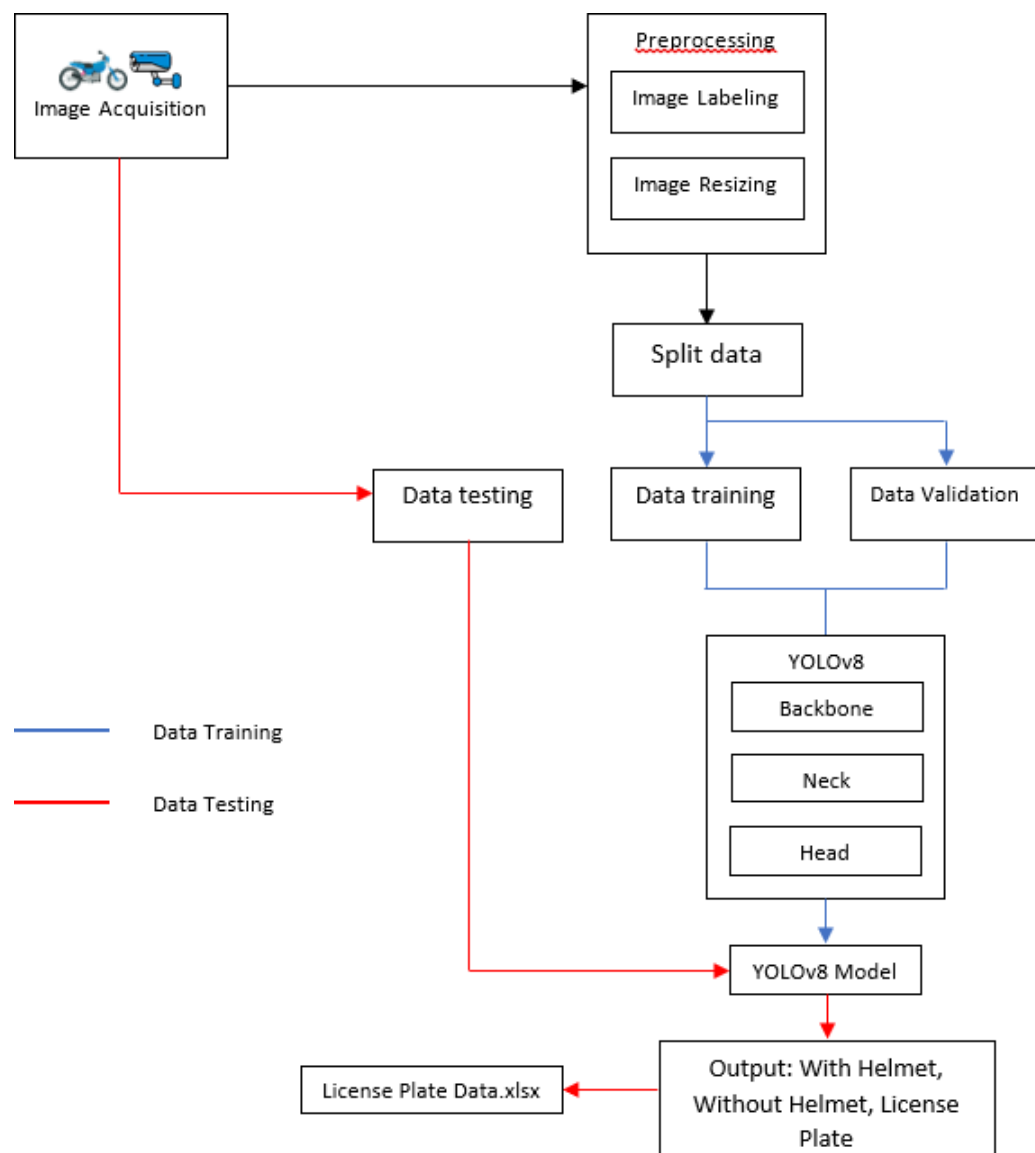
## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini meliputi analisis dan perancangan sistem dalam merancang sistem untuk deteksi helm dan plat sepeda motor. Tahapan yang akan dilakukan mengenai pengambilan data, preprocessing, pendeteksian menggunakan You Only Look Once v8, dan proses training dan testing pada metode You Only Look Once v8.

#### **3.1. Analisis Sistem**

Arsitektur umum dari penelitian ini dapat dilihat dari Gambar 3.1. Untuk mendeteksi helm, dan nomor plat sepeda motor diperlukan beberapa tahapan. Adapun data dibagi menjadi tiga bagian yaitu data training, data validation, dan data testing dimana data training dan validation harus melewati tahap preprocessing sedangkan data testing langsung diuji menggunakan model yang didapat dari hasil training. Untuk tahapan pertama yaitu pengambilan citra dari jalan raya untuk dilakukan preprocessing dimana akan dilakukan resizing agar citra yang digunakan untuk proses training memiliki ukuran yang sama dan menjadi ukuran yang dibutuhkan. Kemudian dilakukan proses labeling untuk mengenali bagian bagian pada helm, dan plat sepeda motor yang akan dideteksi. Selanjutnya You Only Look Once v8 (YOLOv8) akan digunakan untuk mengidentifikasi pengendara sepeda motor, yang bukan pengendara sepeda motor, helm, dan yang tidak menggunakan helm, dan plat sepeda motor yang selanjutnya akan didapatkan bounding box dari objek yang terdeteksi, dengan nama dari objek yang terdeteksi dan *confidence level* dari objek tersebut. Output yang akan dihasilkan dari proses ini adalah pengendara, pengendara dengan helm, pengendara tanpa helm, nomor plat sepeda motor, dan file excel yang berisi nomor plat pengendara sepeda motor yang tidak menggunakan helm.



**Gambar 3. 1** Arsitektur Umum

### 3.2. Image Acquisition

Image Acquisition merupakan tahap pengambilan video yang akan digunakan untuk proses training, validation dan testing. Data video untuk deteksi pengendara dan helm dibagi menjadi 1 frame per detik dengan total jumlah frame setelah dibagi adalah sebanyak 2800 dan berisikan 715 pengendara yang menggunakan helm, dan 829 pengendara yang tidak menggunakan helm. Sedangkan video untuk deteksi pengendara

dibagi menjadi 1 frame per 1,2 detik dengan total 1933 pengendara sepeda motor. Selanjutnya data untuk deteksi plat sepeda motor didapat dari roboflow dengan jumlah 2750 data untuk training dan 452 data untuk validation. Lokasi untuk pengambilan video adalah di Jalan Kapten Sumarsono.

Kamera yang digunakan untuk pengambilan video adalah kamera smartphone POCO F4 64mP dan video diambil dengan resolusi 1080p dan 60 FPS dengan ratio 16:9 dan menggunakan tripod dengan tinggi 160cm.

### **3.3. Preprocessing**

Data video yang diambil akan dibagi menjadi frame, setelah dibagi tahap selanjutnya yang dilakukan adalah preprocessing, dalam tahap ini akan dilakukan resizing dan labeling.

#### **3.3.1. Resizing**

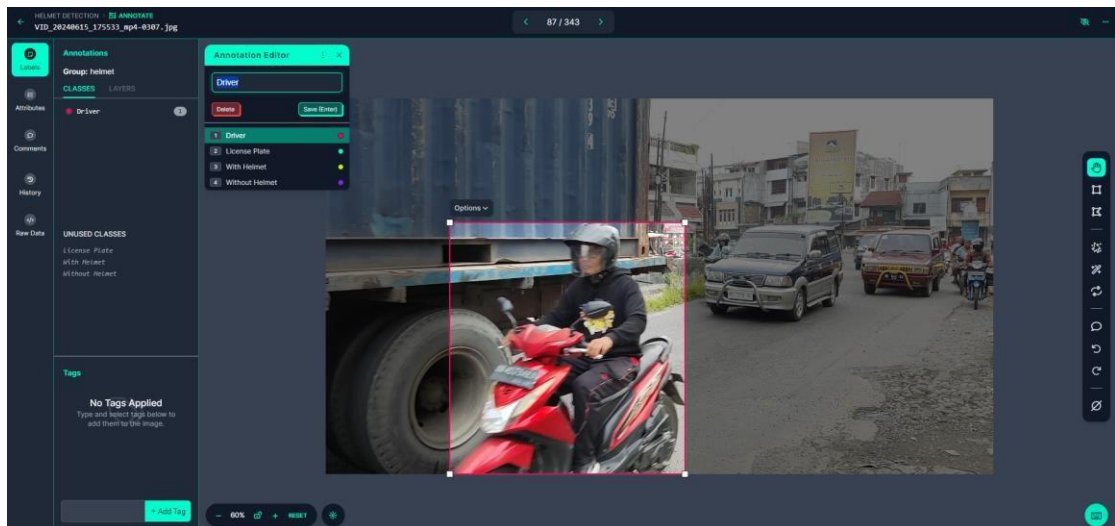
Tahap resizing dilakukan untuk beberapa tujuan utama seperti optimalisasi kinerja, konsistensi, dan standarisasi data, serta efisiensi penyimpanan dan transfer data. Dalam konteks pengolahan gambar, resising dapat mengurangi ukuran file, sehingga mempercepat waktu pemrosesan dan menghemat penggunaan memori, ini sangat penting untuk deteksi objek secara *real time*. Proses resizing dilakukan dengan platform roboflow setelah dilabel dan sebelum di download.

#### **3.3.2. Labeling**

Tahap labeling bertujuan untuk memberi label pada objek yang akan ditraining. Pada penelitian ini digunakan platform roboflow untuk labeling helm, pengendara yang tidak menggunakan helm, pengendara, dan plat sepeda motor. Setelah labeling hasilnya akan disimpan dalam file txt.

Proses labeling adalah tahap yang sangat penting, proses ini menentukan kualitas data yang akan digunakan untuk melatih model machine learning. Label yang akurat dan konsisten memastikan bahwa model dapat belajar dan membuat prediksi dengan baik, yang pada akhirnya menghasilkan aplikasi yang lebih efektif dan efisien.

Cara memberi label pada dataset dapat dilihat dari Gambar 3.2



**Gambar 3. 2** Cara Labeling

### 3.4. Model You Only Look Once v8 (YOLOv8)

Setelah tahap preprocessing, dataset akan diproses dengan metode YOLOv8. Sebelum dilakukan proses training, akan ditentukan beberapa parameter, yaitu jenis model YOLOv8 yang akan digunakan, dan hyperparameter lain.

#### 3.4.1. Clone repository

Pada tahap ini dilakukan proses clone repository dari github ultralytics

#### 3.4.2. Install requirements

Pada tahap ini akan dilakukan instalasi packages atau library yang diperlukan untuk membuat aplikasi yaitu pytorch, CUDA, pandas, opencv, flask, numpy, dan easyocr.

#### 3.4.3. Menambahkan data.yaml

Data.yaml berisi nama class yang akan dideteksi, jumlah class yang akan dideteksi oleh sistem yang akan dibuat, direktori folder untuk training, validation, dan testing.

#### 3.4.4. Proses training

Data yang didapatkan berjumlah 6033 Yang terdiri dari 715 data with helmet dan 829 data without helmet, 1933 data pengendara, dan 5226 data nomor plat sepeda motor dengan data ini maka jumlah epoch yang digunakan untuk training adalah 50 epoch untuk semua data, dan batch size 8 untuk semua data. Model YOLOv8 yang digunakan untuk proses training adalah YOLOv8s.

Arsitektur YOLOv8 terdiri dari backbone yang mengekstraksi fitur dasar dari gambar dan head yang melakukan prediksi lokasi kotak pembatas (bounding boxes) dan kelas objek. YOLOv8 membagi gambar input menjadi grid cells, di mana setiap cell bertanggung jawab mendeteksi objek yang pusatnya berada di dalam cell tersebut. Dengan prediksi multiskala, setiap grid cell menghasilkan beberapa prediksi kotak pembatas dan kelas objek, memungkinkan deteksi objek dari berbagai ukuran. YOLOv8 menghasilkan bounding boxes dengan koordinat yang relatif terhadap grid cell, disertai dengan skor kepercayaan yang menunjukkan seberapa yakin model bahwa kotak tersebut mengandung objek, serta skor kelas untuk setiap objek yang mungkin.

Setelah menghasilkan bounding box, YOLOv8 menggunakan algoritma Non-Maximum Suppression (NMS) untuk mengeliminasi kotak yang tumpang tindih dan memilih kotak dengan skor kepercayaan tertinggi, sehingga setiap objek hanya terdeteksi sekali. Kotak pembatas dengan skor kepercayaan di bawah ambang batas tertentu diabaikan untuk mengurangi deteksi palsu. YOLOv8 menggunakan fungsi loss yang menggabungkan beberapa komponen seperti loss untuk kotak pembatas, loss untuk klasifikasi objek, dan loss untuk kepercayaan, memastikan model dapat belajar mendeteksi dan mengklasifikasikan objek dengan akurasi tinggi.

Proses pelatihan melibatkan penyesuaian bobot jaringan berdasarkan fungsi loss untuk meminimalkan kesalahan prediksi. Setelah dilatih, model dapat digunakan untuk mendeteksi objek dalam gambar atau video baru secara waktu nyata, menghasilkan kotak pembatas dan label kelas untuk setiap objek yang terdeteksi. Secara keseluruhan, YOLOv8 menggabungkan kecepatan dan akurasi tinggi, menjadikannya pilihan populer untuk berbagai aplikasi seperti pengawasan video, sistem keamanan, dan analisis gambar.

Dibawah ini adalah pseudocode untuk proses training

```
# Import libraries
import ultralytics
import torch
```

```

# Set device to CUDA

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

# Set dataset directory
dataset_directory = "/path/to/dataset"

# Load YOLOv8 model
model = YOLO('yolov8s')

# Set hyperparameters
epochs = 50
batch_size = 8
data_yaml = "path/to/data.yaml"

# Function to train the model
def train(model, data_yaml, epochs, batch_size):
    # Load dataset
    dataset = load_dataset(data_yaml)

    # Training loop
    for epoch in range(epochs):
        # Get batches of data
        for batch in get_batches(dataset, batch_size):
            # Forward pass
            predictions = model.forward(batch['inputs'])

            # Compute loss
            loss = compute_loss(predictions, batch['targets'])

            # Backward pass and update weights
            model.backward(loss)
            update_weights(model)

    # Validate the model
    validate(model, dataset['validation'])

```

```

# Log metrics
log_metrics(epoch, loss, accuracy)

# Save checkpoint
save_checkpoint(model, epoch)

# Train the model
train(model, data_yaml, epochs, batch_size)

```

#### 3.4.4. Proses testing

Data yang digunakan untuk proses testing diambil terpisah dengan durasi 7 menit dimana data tersebut berisi 309 pengendara sepeda motor, 285 pengendara dengan helm, 277 pengendara tanpa helm, dan 293 plat sepeda motor. Dan proses ini dilakukan menggunakan model yang telah di train sebelumnya. Model tersebut digunakan untuk melakukan deteksi helm, pengendara, dan plat sepeda motor tanpa ditraining ulang. Tujuan dari proses ini adalah untuk menguji apakah sistem yang dikembangkan dapat mendeteksi pemakaian helm pada pengendara sepeda motor dan mengekstraksi plat pengendara yang tidak memakai helm.

Pseudocode berikut menunjukkan tahapan dari proses tesing yang dilakukan

```

# Import necessary libraries

import pandas as pd

import os

import cv2

from flask import Flask, render_template, request, redirect, url_for

from werkzeug.utils import secure_filename

from ultralytics import YOLO

# Initialize Flask app

app = Flask(__name__)

# Set upload folder

UPLOAD_FOLDER = '/path/to/upload'

```

```

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# Allowed file extensions
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif', 'mp4'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

# Load YOLOv8 models
model_helmet = YOLO('path/to/helmet_model.pt')
model_plate = YOLO('path/to/plate_model.pt')

@app.route('/', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        # Check if a file is in the request
        if 'file' not in request.files:
            return redirect(request.url)

        file = request.files['file']

        if file.filename == "":
            return redirect(request.url)

        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))

            if filename.rsplit('.', 1)[1].lower() == 'mp4':
                detect_object_in_video(os.path.join(app.config['UPLOAD_FOLDER'],
filename))
            else:
                detect_object_in_image(os.path.join(app.config['UPLOAD_FOLDER'],
filename))

            return render_template('upload.html', filename=filename)

    return render_template('upload.html')

```



```

def detect_object_in_image(image_path):
    # Load image
    image = cv2.imread(image_path)

    if image is None:
        print("Error loading image")
        return

    # Detect motorcycles
    motorcycles = model_helmet(image)

    for motorcycle in motorcycles:
        # Extract region of interest
        roi = extract_roi(image, motorcycle)

        # Detect helmet
        helmet_detection = model_helmet(roi)

        if 'without helmet' in helmet_detection:
            # Detect license plate
            plates = model_plate(roi)

            for plate in plates:
                # Perform OCR on the license plate
                plate_text = perform_ocr(plate)
                save_to_excel(plate_text)
            save_output_image(image)

def detect_object_in_video(video_path):
    cap = cv2.VideoCapture(video_path)
    out = None

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

```

```

# Detect motorcycles
motorcycles = model_helmet(frame)

for motorcycle in motorcycles:
    # Extract region of interest
    roi = extract_roi(frame, motorcycle)

    # Detect helmet
    helmet_detection = model_helmet(roi)

    if 'without helmet' in helmet_detection:
        # Detect license plate
        plates = model_plate(roi)

        for plate in plates:
            # Perform OCR on the license plate
            plate_text = perform_ocr(plate)
            save_to_excel(plate_text)

if out is None:
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('output.avi', fourcc, 20.0, (frame.shape[1],
rame.shape[0]))

    out.write(frame)

cap.release()
out.release()

if __name__ == '__main__':
    app.run(debug=True)

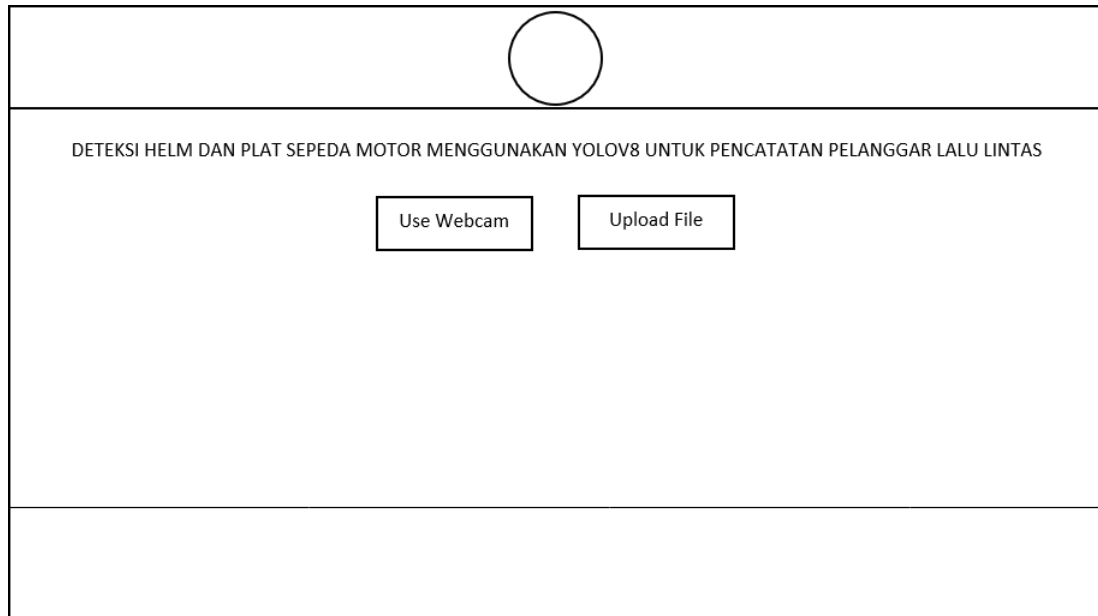
```

### 3.5. Perancangan Antarmuka Sistem

Perancangan antarmuka dalam sistem deteksi helm ini bertujuan untuk menyediakan pengalaman pengguna yang intuitif dan efisien dalam menggunakan sistem yang dibangun.

#### 3.5.1. Rancangan Tampilan Utama

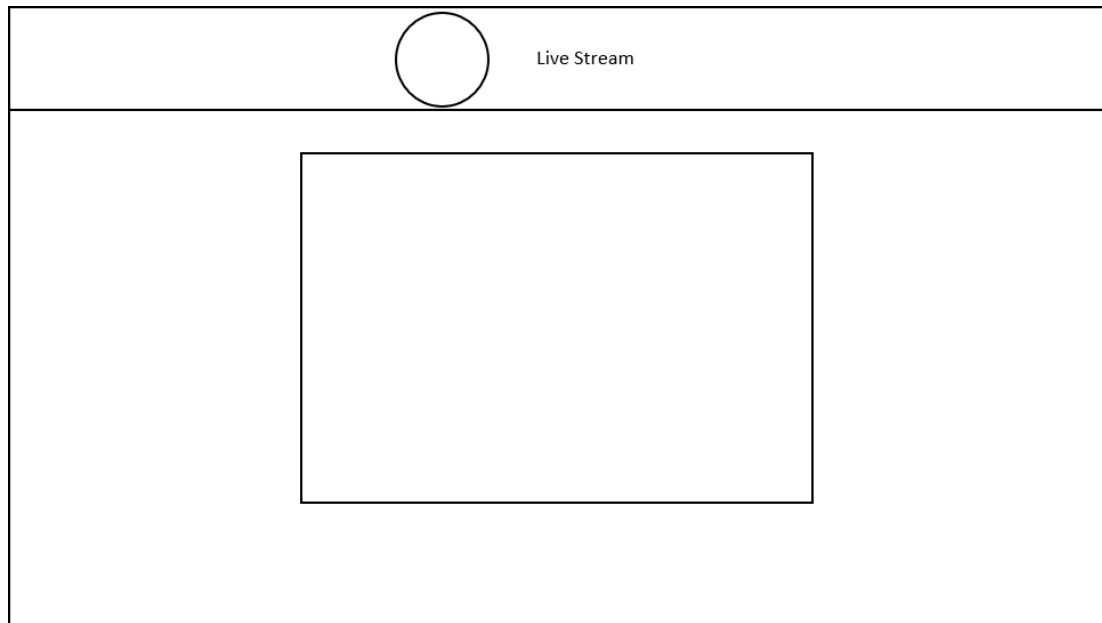
Pada halaman awal, terdapat dua pilihan untuk menggunakan aplikasi ini yaitu menggunakan webcam untuk mendeteksi secara realtime dan bisa juga menggunakan video yang sudah tersimpan di komputer. Rancangan halaman awal dapat dilihat pada Gambar 3.3



**Gambar 3. 3** Rancangan Tampilan Utama

### 3.5.2. Rancangan Tampilan Live Stream

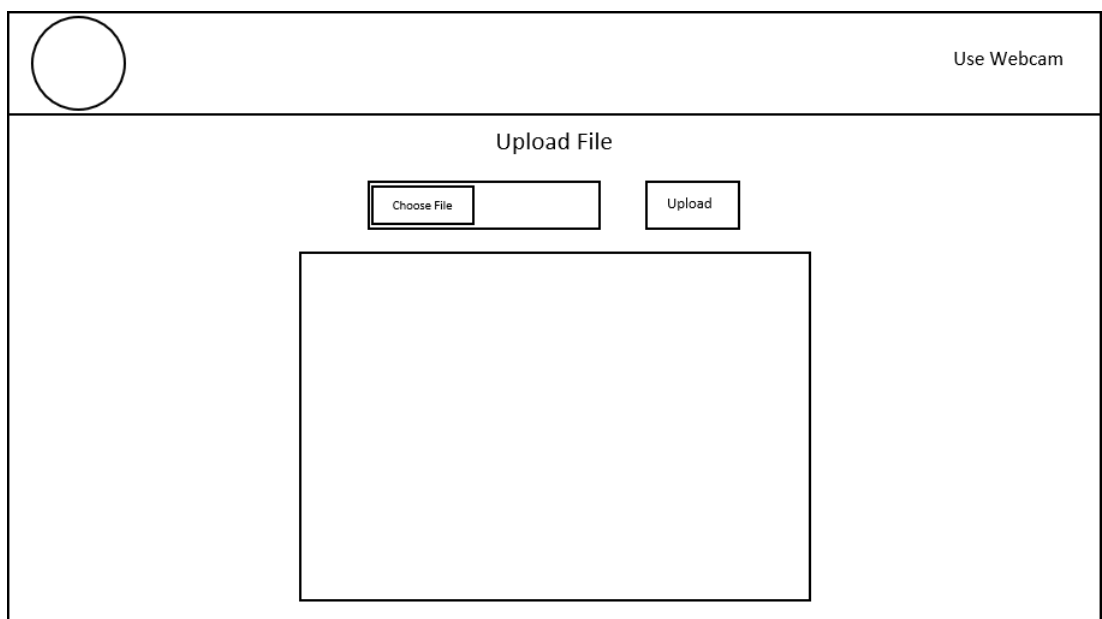
Jika pengguna memilih opsi use webcam, maka akan diarahkan ke halaman live stream, dimana di halaman ini pengguna bisa menggunakan webcam untuk melakukan deteksi secara langsung. Rancangan tampilan live stream dapat dilihat pada Gambar 3.4



**Gambar 3. 4** Rancangan Tampilan Live Stream

### 3.5.3. Rancangan Tampilan upload video

Pada halaman ini, pengguna dapat mengunggah video yang ingin dideteksi. Halaman tersebut berisi sebuah form untuk mengunggah video dan menampilkan hasil deteksi video yang sudah diunggah. Rancangan tampilan untuk opsi mengunggah video dapat dilihat pada Gambar 3.5



**Gambar 3. 5** Rancangan Tampilan Upload File

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN**

Bab ini mengulas penerapan metode You Only Look Once version 8 untuk mendeteksi helm dan plat sepeda motor berdasarkan sistem yang telah dirancang pada bab sebelumnya.

#### **4.1. Implementasi Sistem**

Pada tahap ini penerapan metode YOLOv8 untuk deteksi helm dan plat sepeda motor akan dilakukan dalam sebuah sistem. Sistem dibangun dengan menggunakan Bahasa pemrograman Python, dan berbasis website dengan memanfaatkan HTML CSS serta menggunakan framework Flask.

##### **4.1.1. Perangkat keras dan lunak yang digunakan**

Perangkat keras dan lunak yang digunakan untuk membuat sistem untuk mendeteksi helm dan plat sepeda motor adalah sebagai berikut:

1. CPU : 12th Gen Intel(R) Core(TM) i7-12700F 2.10 GHz
2. GPU : RTX 3060 (include CUDA)
3. RAM : 32GB, DDR4
4. Kapasitas HDD 1TB, dan SSD 500GB
5. OS : Windows 11 Pro 64-bit

Implementasi sistem ini dilakukan dengan menggunakan Bahasa pemrograman Python 3.10 dengan framework Flask dan beberapa library pendukung seperti:

1. Flask
2. OpenCV
3. Numpy
4. Easyocr
5. Pandas

#### 4.1.2. Implementasi Rancangan Antar Muka

Setelah merancang antarmuka pada bab 3, langkah selanjutnya adalah mengimplementasikan rancangan tersebut dimana rancangan tersebut berisikan tampilan utama, tombol untuk menggunakan webcam, dan tombol untuk mengunggah file.

##### 4.1.2.1. Tampilan Utama

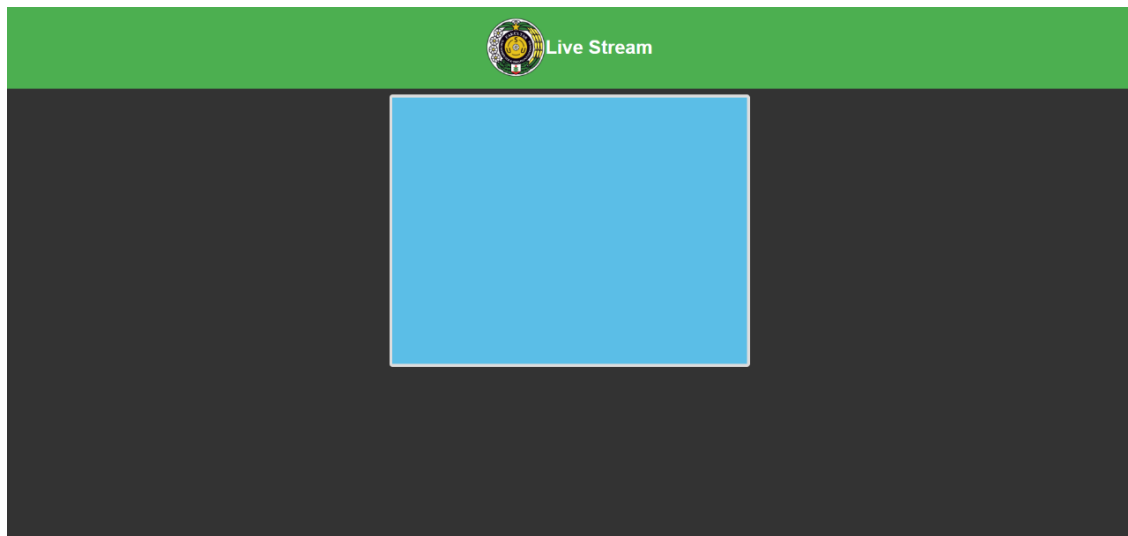
Tampilan utama ini akan muncul saat pertama kali membuka aplikasi, tampilan ini berfungsi sebagai halaman awal. Pada halaman ini, terdapat dua tombol yaitu tombol use webcam yang akan mengarahkan pengguna ke halaman live stream, dan tombol upload file yang akan membawa pengguna ke halaman upload file. Tampilan utama dapat dilihat pada Gambar 4.1



**Gambar 4. 1** Tampilan *Utama*

##### 4.1.2.2. Tampilan Live Stream

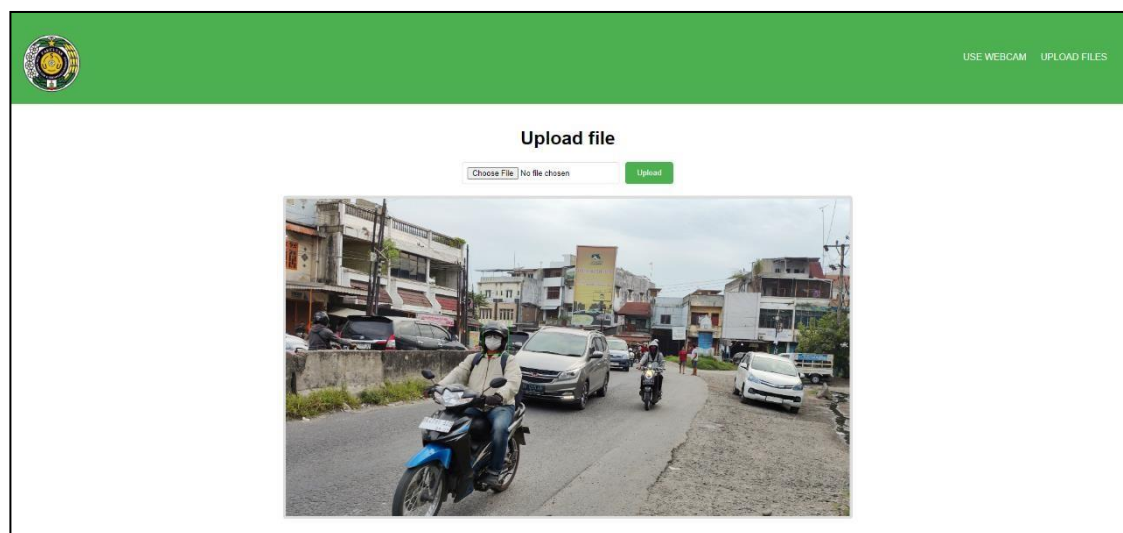
Halaman ini digunakan untuk mendeteksi helm dan plat sepeda motor menggunakan webcam dimana proses pendeteksian ini dilakukan secara realtime. Rancangan tampilan live stream dapat dilihat pada Gambar 4.2



**Gambar 4. 2** Tampilan Live Stream

#### 4.1.2.3. Tampilan Upload File

Halaman ini digunakan untuk mendeteksi helm dan plat sepeda motor menggunakan video yang sudah tersimpan di computer pengguna. Rancangan tampilan upload fiile dapat dilihat pada Gambar 4.3



**Gambar 4. 3** Tampilan Upload File

#### 4.1.3. Implementasi Data

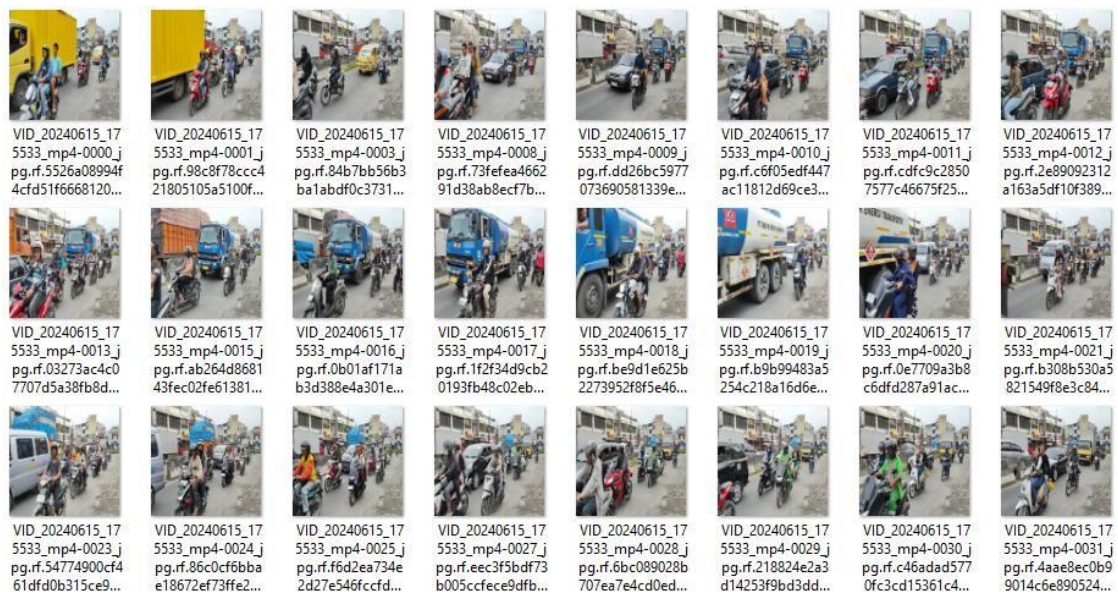
Data diambil di Jalan Kapten Sumarsono dimana data tersebut diambil selama dua hari diambil di waktu siang selama satu jam per hari. Kamera yang digunakan untuk mengambil data adalah kamera smartphone POCO F4 dengan resolusi 1920 x 1080

64MP 60 FPS. Jumlah dataset yang didapat dari data tersebut adalah 890 dataset untuk training juga validation model deteksi helm dan tanpa helm, dan 917 dataset untuk training serta validation model deteksi pengendara sepeda motor. Dari data tersebut terdapat 715 pengendara yang menggunakan helm, 829 pengendara yang tidak menggunakan helm, dan 1933 pengendara sepeda motor. Data yang digunakan untuk proses testing diambil selama tujuh menit di Jalan Kapten Sumarsono terpisah dari data training, dan data validation. Dataset yang digunakan untuk training dan validation dari class helmet dan validation dapat dilihat pada Gambar 4.4, sedangkan dataset untuk training dan validation dari class driver dapat dilihat pada Gambar 4.5.



**Gambar 4. 4** Data training dan Validation Helmet





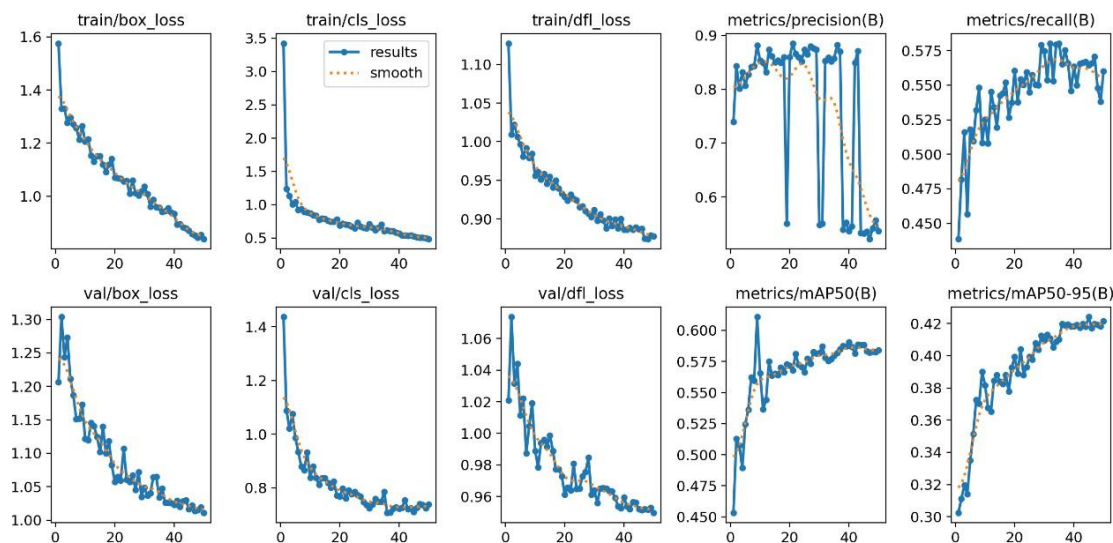
**Gambar 4. 5** Data training dan Validation Driver

## 4.2. Training model YOLOv8

Untuk training model YOLOv8 ada beberapa parameter yang harus ditentukan, diantaranya epoch, dan batch size. Jumlah epoch yang digunakan dalam penelitian ini adalah 50 untuk training model deteksi helm, plat sepeda motor dan pengendara sepeda motor. Kemudian batch size yang digunakan untuk training model deteksi helm, plat sepeda motor, dan pengendara sepeda motor adalah 8. Berikut adalah hasil training model deteksi helm, plat sepeda motor, dan pengendara sepeda motor.

### 4.2.1. Hasil training model deteksi helm

Waktu yang dibutuhkan untuk training model deteksi helm sebanyak 50 epoch adalah sekitar 10 menit. Nilai recall yang didapat dari training ini adalah 0.56 dimana nilai recall mengalami peningkatan dari nilai awal yaitu 0.43. Nilai mAP 50 yang didapat juga meningkat yang awalnya 0.45 menjadi 0.58, dan juga mAP50-95 dengan nilai awal 0.3 dan nilai akhir 0.42. Hasil training model deteksi helm dapat dilihat di Gambar 4.6



**Gambar 4. 6** Hasil training model deteksi helm

Pada Gambar 4.6 terlihat bahwa nilai box loss, cls\_loss (Classification Loss), dan dfl\_loss (Validation Distribution Focal Loss) menurun seiring dengan bertambahnya epoch, yang berarti model semakin baik dalam memprediksi bounding box untuk objek yang terdeteksi pada training. Sementara itu precision mengalami fluktuasi yang signifikan dan cenderung menurun pada akhir proses training, hal ini disebabkan oleh adanya satu class yang tidak memiliki data sama sekali yang menyebabkan ketidakseimbangan jumlah data sehingga model tidak terlatih dengan baik untuk mendeteksi class tersebut.

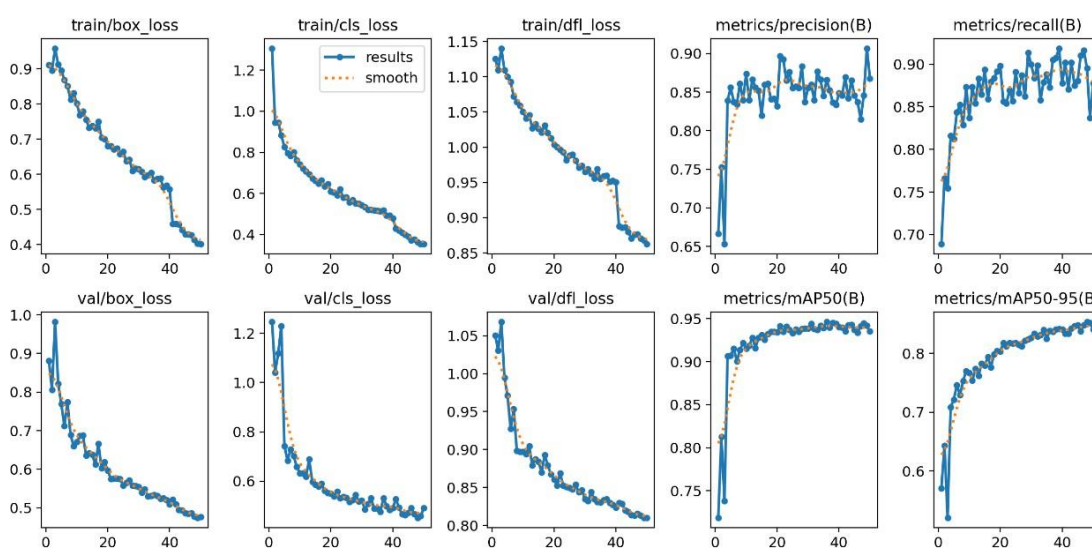
Recall yang didapat setelah proses training cenderung meningkat seiring bertambahnya epoch yang menunjukkan peningkatan kemampuan model dalam mendeteksi objek. Nilai val/box\_loss (box loss pada data validasi) menurun menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada data yang tidak terlihat selama proses training, nilai val/cls\_loss atau classification loss pada data validasi menunjukkan bahwa performa model dalam mengklasifikasikan objek pada data validation mengalami peningkatan, dan val/dfl\_loss (DFL loss pada data validasi), juga menurun seiring bertambahnya epoch, menunjukkan bahwa model mampu melakukan generalisasi dalam memprediksi distribusi nilai pada data yang tidak pernah ditampilkan selama proses training.

Hasil proses training ini menunjukkan bahwa nilai mAP50 mengalami peningkatan menunjukkan peningkatan performa model dalam mendeteksi objek

dengan presisi tinggi pada data validation. Sementara itu nilai mAP 50-59 juga menunjukkan peningkatan yang berarti model semakin baik dalam mendeteksi objek pada berbagai ukuran dan kondisi.

#### 4.2.2. Hasil Training model deteksi pengendara sepeda motor

Waktu yang dibutuhkan untuk training model deteksi pengendara sepeda motor juga sekitar 10 menit. Nilai precision dari training ini adalah 0.85, mAP50 yang didapat sebesar 0.93, dan mAP50-95 sebesar 0.84. Hasil training model deteksi pengendara sepeda motor ditunjukkan pada Gambar 4.7



**Gambar 4. 7** Hasil training model deteksi pengendara sepeda motor

Pada Gambar 4.7 terlihat bahwa train/box\_loss mengalami penurunan selama proses pelatihan, mengindikasikan bahwa model semakin baik dalam memprediksi bounding box dari objek yang terdeteksi. train/cls\_loss menunjukkan penurunan nilai dari classification loss selama proses pelatihan, yang berarti model semakin baik dalam mengklasifikasikan objek yang terdeteksi. train/dfl\_loss juga menunjukkan penurunan, yang menunjukkan bahwa model semakin baik dalam memprediksi distribusi nilai yang sesuai dengan objek yang dideteksi. Precision pada gambar tersebut relatif stabil, menunjukkan bahwa model dapat memprediksi bounding boxes dengan benar. Recall menunjukkan peningkatan selama proses pelatihan, yang menandakan bahwa model semakin baik dalam menemukan 'True Positive' dari kelas tertentu.

Nilai val/box\_loss menunjukkan penurunan selama validasi, yang mengindikasikan bahwa model mampu memprediksi bounding box pada data yang tidak terlihat selama proses pelatihan dengan baik. Nilai val/cls\_loss yang menurun menunjukkan bahwa model dapat mengklasifikasikan objek dengan baik pada data validasi. Nilai val/df\_l\_loss yang menurun menunjukkan bahwa model mampu memprediksi distribusi nilai pada data yang belum pernah dilihat selama proses pelatihan.

metrics/mAP50 pada Gambar 4.7 menunjukkan bahwa model semakin baik dalam mendeteksi objek dengan presisi tinggi pada data validasi. metrics/mAP50-95 menunjukkan peningkatan nilai average precision, mengindikasikan bahwa model semakin baik dalam mendeteksi objek pada berbagai ukuran dan kondisi.

#### 4.2.3. Hasil training model deteksi plat sepeda motor

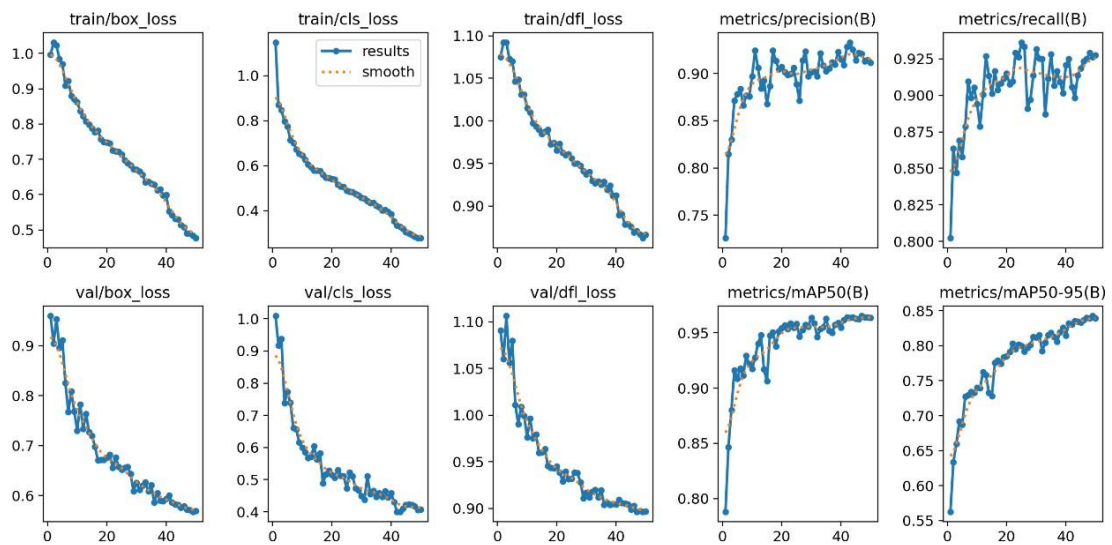
Waktu yang dibutuhkan untuk training model deteksi plat sepeda motor adalah sekitar 1 jam. Nilai precision dari training deteksi plat sepeda motor adalah 0.9, nilai mAP50 sebesar 0.96, dan mAP50-95 sebesar 0.83 Hasil training model deteksi plat sepeda motor terdapat pada Gambar 4.8

Pada Gambar 4.8 terlihat bahwa train/box\_loss mengalami penurunan selama proses pelatihan yang mengindikasikan bahwa model semakin baik dalam memprediksi bounding box dari objek yang terdeteksi. Train/cls\_loss menunjukkan penurunan nilai dari classification loss selama proses training, dan menunjukkan bahwa model semakin baik dalam mengklasifikasikan objek yang terdeteksi. Train/df\_l\_loss juga menunjukkan bahwa model semakin baik dalam memprediksi distribusi nilai yang sesuai dengan objek yang dideteksi. Precision dari gambar tersebut relatif stabil yang menunjukkan bahwa model dapat memprediksi bounding boxes dengan benar. Recall menunjukkan peningkatan selama proses training yang tandanya model semakin baik dalam menemukan 'True Positive' dari class tertentu.

Nilai val/box\_loss menunjukkan penurunan nilai classification loss selama validasi yang menunjukkan bahwa model mampu memprediksi bounding box pada data yang tidak terlihat ada proses training dengan baik. Nilai val/cls\_loss yang menurun menunjukkan model dapat mengklasifikasikan objek dengan baik pada data validation.

Nilai val/dfl\_loss yang mengalami penurunan menunjukkan bahwa model mampu memprediksi distribusi pada data yang belum pernah dilihat saat proses training.

Metric/mAP50 pada Gambar 4.8 menunjukkan bahwa model semakin baik dalam mendeteksi objek dengan presisi tinggi pada data validation. Metric/mAP50\_95 menunjukkan peningkatan nilai average precision menunjukkan bahwa model semakin baik dalam mendeteksi objek pada berbagai ukuran dan kondisi.



**Gambar 4. 8** Hasil training model deteksi plat sepeda motor

Setelah dilakukan proses training, akan dihasilkan dua model yaitu best.pt dan last.pt dan model yang dipilih untuk digunakan dalam sistem yang dibuat adalah best.pt untuk deteksi helm, plat sepeda motor, dan pengendara sepeda motor.

#### 4.3. Prosedur Operasional

Halaman utama aplikasi ini berisi pilihan untuk memilih sumber video yang akan dideteksi. Pada halaman ini terdapat pilihan untuk menggunakan webcam atau upload file foto atau video.

Agar aplikasi ini dapat digunakan, pengguna harus membuka command prompt (CMD) dan set direktori dimana aplikasi ini tersimpan dan jalankan perintah 'flask run'.

```
D:\deteksi>flask run
```

**Gambar 4. 9** Cara Menjalankan Aplikasi

Setelah menjalankan perintah tersebut, akan muncul sebuah alamat. Untuk membuka alamat tersebut bisa disalin dan tempelkan di browser, atau dengan menekan tombol Ctrl + klik alamat tersebut.



```
* Debug mode: off
WARNING: This is a development server. Do not use in production.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
http://127.0.0.1:5000
Ctrl+Click to follow link
```

**Gambar 4. 10** Alamat Lokal Flask

Setelah memasukkan alamat di browser atau klik alamat tersebut maka pengguna akan diarahkan ke halaman awal aplikasi.

#### 4.4. Pengujian Sistem




Dalam tahap pengujian sistem, akan dilakukan pengujian terhadap data yang telah diambil terpisah dari data training dan validation dengan durasi 7 menit 34 detik di Jalan Kapten Sumarsono. Pengujian akan dilakukan menggunakan video tersebut dimana terdapat 308 pengendara sepeda motor, 285 pengendara dengan helm, 277 pengendara tanpa helm. Dan 293 plat sepeda motor.




Berdasarkan hasil pengujian, untuk model deteksi helm mendapat akurasi sebesar 80,28%, recall 0,82%, precision 0,93, dan F1 Score 0,87. Selanjutnya untuk deteksi pengendara didapatkan akurasi sebesar 90,26%, recall 0,90%, precision 1, dan F1 Score 0,94. Untuk deteksi plat sepeda motor akurasi yang dihasilkan adalah 76,8%, recall 0,76, precision 1, dan F1 Score 0,87.

No	Jenis Deteksi	Akurasi	Recall	Precision	F1 Score
1	Deteksi Helm	80,28%	0,82	0,93	0,87
2	Deteksi Pengendara	90,26%	0,90	1	0,94
3	Deteksi Plat Sepeda Motor	76,8%	0,76	1	0,87




**Tabel 4. 1** Tabel Akurasi Proses Training






No	Gambar	Prediksi	Aktual	Status
1		Driver  With Hemet	Driver  With Helmet	Berhasil  Berhasil
2		Driver  Without Helmet	Driver  Without Helmet	Berhasil  Berhasil
3		Driver  With Helmet	Driver  With Helmet	Berhasil  Berhasil




4		Driver  With Helmet	Driver  With Helmet	Berhasil  Berhasil
5		Driver  Without Helmet  License Plate	Driver  Without Helmet  License Plate	Berhasil  Berhasil  Berhaisl
6		Driver  Helmet	Driver  Helmet	Berhasil  Berhasil






7		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		License Plate	License Plate	Berhasil
8		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		Without Helmet	Without Helmet	Berhasil
		License Plate	License Plate	Berhasil
9		Driver	Driver	Berhasil
		With Helmet	With Helmet	Berhasil


10		Driver  With Helmet	Driver  Without Helmet	Berhasil  Gagal
11		Driver  Without Helmet  -  License Plate	Driver  Without Helmet  Without Helmet  License Plate	Berhasil  Berhasil  Gagal  Berhasil
12		Driver  With Helmet  Without Helmet  -	Driver  With Helmet  Without Helmet  License Plate	Berhasil  Berhasil  Berhasil  Gagal




13		Driver	Driver	Berhasil
		Driver	Driver	Berhasil
		With Helmet	With Helmet	Berhasil
		With Helmet	With Helmet	Berhasil
14		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		License Plate	License Plate	Berhasil
15		Driver	Driver	Berhasil
		With Helmet	With Helmet	Berhasil
		Without Helmet	Without Helmet	Berhasil
		-	License Plate	Gagal




16		Driver Driver	Driver Driver	Berhasil Berhasil
		With Helmet -	With Helmet With Helmet	Berhasil Gagal
		Without Helmet	Berhasil	Berhasil
		-	License Plate	Gagal
17		-  -	Driver  With Helmet	Gagal  Gagal
18		Driver  Without Helmet  Without Helmet  License Plate	Driver  Without Helmet  Without Helmet  License Plate	Berhasil  Berhasil  Berhasl  Berhasil

19		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		Without Helmet	Without Helmet	Berhasil
		License Plate	License Plate	Berhasil
20		Driver	Driver	Berhasil
		-	With Helmet	Gagal
21		Driver	Driver	Berhasil
		Without Helmet	Without Hemlet	Berhasil
		License Plate	License Plate	Berhasil



22		Driver  With Helmet  -	Driver  With Helmet  Without Helmet	Berhasil  Berhasil  Gagal
23		Driver  With Helmet	Driver  With Helmet	Berhasil  Berhasil
24		Driver  With Helmet  Without Helmet  -	Driver  With Helmet  Without Helmet  License Plate	Berhasil  Berhasil  Berhasil  Gagal

25		Driver  With Helmet  With Helmet	Driver  With Helmet  With Helmet	Berhasil  Berhasil  Berhasil
26		Driver  Without Helmet  License Plate	Driver  Without Helmet  License Plate	Berhasil  Berhasil  Berhasil
27		Driver  Without Helmet  Without Helmet	Driver  Without Helmet  Without Helmet	Berhasil  Berhasil  Berhasil

28		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		Without Helmet	Without Helmet	Berhasil
29		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		Without Helmet	Without Helmet	Berhasil
		License Plate	License Plate	Berhasil
30		Driver	Driver	Berhasil
		Without Helmet	Without Helmet	Berhasil
		License Plate	License Plate	Berhasil

Tabel 4. 2 Hasil Pengujian

Setelah dilakukan pengujian sistem menggunakan metode YOLOv8, terdapat beberapa kesalahan deteksi dan gambar helm pengendara sepeda motor, dan pengendara tanpa helm yang tidak terdeteksi. Faktor yang mempengaruhi kesalahan tersebut diantaranya bounding box 'Driver' yang tidak terdeteksi karena terhalang kendaraan lain,



penumpang yang bersembunyi dibalik pengendara, dan mungkin juga bisa disebabkan kurangnya variasi data yang digunakan untuk training. Faktor yang mempengaruhi kesalahan dalam deteksi plat sepeda motor adalah bounding box ‘driver’ yang terhalang kendaraan lain, dan plat sepeda motor yang dideteksi ada yang miring ke atas ke kanan atau ke kiri, dan ada pengendara tanpa helm yang terdeteksi menggunakan helm sehingga tidak dilakukan deteksi plat oleh sistem. Selanjutnya factor yang mempengaruhi kesalahan deteksi pengendara adalah pengendara tersebut terhalang oleh pengendara atau kendaraan lain.

	TP	FP	FN
With Helmet	156	10	33
Without Helmet	121	2	25
Driver	278	0	30
License Plate	80	0	24
Total	630	10	112

**Tabel 4. 3** Nilai True Positive, False Positive, dan False Negative

Berdasarkan table 4.3 TP dari class With Helmet adalah ketika model memprediksi adanya helm pada pengendara, dan prediksi tersebut benar. FP dari class With Helmet adalah Ketika model memprediksi adanya helm pada pengendara namun prediksi tersebut salah. FN dari class With Helmet adalah Ketika model memprediksi tidak ada helm pada pengendara, namun prediksi tersebut salah.

TP pada class Without Helmet adalah ketika model memprediksi adanya pengendara yang tidak memakai helm, dan prediksi tersebut benar. FP dari class Without Helmet adalah ketika model memprediksi adanya pengendara tanpa helm dan prediksi tersebut salah. FN dari class Without Helmet adalah Ketika model memprediksi tidak ada pengendara tanpa helm, namun prediksi tersebut salah.

TP pada class Driver adalah Ketika model memprediksi adanya pengendara dan prediksi tersebut benar. FP dari class Driver adalah Ketika model memprediksi adanya pengendara namun prediksi tersebut salah. FN dari class Driver adalah Ketika model memprediksi tidak ada pengendara, namun prediksi tersebut salah.

TP pada class License Plate adalah Ketika model memprediksi adanya plat dan prediksi tersebut benar. FP adalah Ketika model memprediksi adanya plat namun

prediksi tersebut salah. FN pada class License Plate adalah Ketika model memprediksi tidak adanya plat namun prediksi tersebut salah.

Untuk menentukan akurasi dari penelitian ini dapat dilakukan dengan cara:

$$\text{Akurasi} = \frac{630}{630+10+112} = 83\%$$

Setelah dilakukan deteksi plat sepeda motor, langkah selanjutnya adalah mengekstraksi plat sepeda motor yang terdeteksi. Dalam penelitian ini, total plat sepeda motor yang terdeteksi sebanyak 80 sedangkan yang bisa diekstraksi hanya 22 plat sepeda motor. Hal ini disebabkan oleh beberapa faktor diantaranya ada cahaya berlebih yang menyebabkan plat sepeda motor tidak terlihat, plat yang menghadap atas, kanan atau kiri dan gambar plat sepeda motor yang blur.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Dalam penelitian ini, telah dilakukan implementasi metode YOLOv8 untuk mendeteksi helm dan plat sepeda motor. Sistem ini dibangun dengan bahasa pemrograman Python dengan framework Flask. Uji coba yang dilakukan menunjukkan bahwa sistem yang telah dibuat mampu mendeteksi helm dengan akurasi 80,28%, deteksi pengendara dengan akurasi 90,26%, dan deteksi plat sepeda motor dengan akurasi 76,8%. Namun akurasi yang didapat untuk ekstraksi plat sepeda motor hanya 27,5%.

Secara keseluruhan, sistem ini telah berfungsi dengan baik dalam mendeteksi helm, pengendara sepeda motor, dan plat sepeda motor, tetapi masih memerlukan peningkatan lebih lanjut terutama dalam ekstraksi plat sepeda motor yang akurasinya masih rendah.

#### **5.2. Saran**

Berikut adalah saran yang diharapkan dapat meningkatkan performa sistem yang akan dikembangkan selanjutnya:

1. Untuk proses training, gunakan jumlah dataset yang lebih banyak dan lebih bervariasi.
2. Saat mengambil data gunakan kamera khusus untuk merekam kondisi jalan agar bisa mendapat resolusi yang lebih baik.
3. Saat merekam usahakan agar posisi plat tidak miring untuk meningkatkan performa OCR.
4. Menguji sistem dalam berbagai kondisi lingkungan seperti kondisi cahaya rendah, cuaca buruk untuk memastikan sistem dapat bekerja dalam berbagai situasi.

## DAFTAR PUSTAKA

- Darmawan H. (2022). *IMPLEMENTASI METODE YOU ONLY LEARN ONE REPRESENTATION*.
- Gao, M., Du, Y., Yang, Y., & Zhang, J. (2019). Adaptive anchor box mechanism to improve the accuracy in the object detection system. *Multimedia Tools and Applications*, 78(19), 27383–27402. <https://doi.org/10.1007/s11042-019-07858-w>
- Goodfellow, I., Bengio, Y., & Courville, A. (n.d.). *Deep Learning*.
- Gunadi, K., & Setyati, E. (2020). *Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network*.
- Kulkarni, M. V., Pawar, D., Talwekar, S., Bharambe, R., & Mahadik, A. (2023). *Helmet, Number Plate Detection and Stolen vehicle recognition using Machine Learning* (Vol. 11). [www.ijcrt.org](http://www.ijcrt.org)
- Lin, H., Deng, J. D., Albers, D., & Siebert, F. W. (2020). Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning. *IEEE Access*, 8, 162073–162084. <https://doi.org/10.1109/ACCESS.2020.3021357>
- Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors*, 22(2). <https://doi.org/10.3390/s22020464>
- Padilla, R., Netto, S. L., Da Silva, E. A. B., & Netto, S. L. (n.d.). *A Survey on Performance Metrics for Object-Detection Algorithms*. <https://doi.org/10.1109/IWSSIP48289.2020>
- Pan, S., Liu, J., & Chen, D. (2022). Research on License Plate Detection and Recognition System based on YOLOv7 and LPRNet. In *Academic Journal of Science and Technology* (Vol. 4, Issue 2).
- Raras, C., & Widiawati, A. (2019). *JISA (Jurnal Informatika dan Sains) Automatic RoI dan Active Contour untuk Deteksi Penggunaan Helm pada Pengendara Sepeda Motor*. 02(02).
- Rosenfeld, A. (1988). An overview of computer vision. In *Artificial Intelligence VI* (Vol. 937). <http://proceedings.spiedigitallibrary.org/>

- Shinde, S., Kothari, A., & Gupta, V. (2018). YOLO based Human Action Recognition and Localization. *Procedia Computer Science*, 133, 831–838. <https://doi.org/10.1016/j.procs.2018.07.112>
- Talib, M., Al-Noori, A. H. Y., & Suad, J. (2024). YOLOv8-CAB: Improved YOLOv8 for Real-time Object Detection. *Karbala International Journal of Modern Science*, 10(1), 56–68. <https://doi.org/10.33640/2405-609X.3339>
- Yanto, Azis Faruq, & Irmawati. (2023). *YOLO-V8 PENINGKATAN ALGORITMA UNTUK DETEKSI PEMAKAIAN MASKER WAJAH*.
- Y. Jamtsho, P. Riyamongkol, and R. Waranusast, “Real-time license plate detection for non-helmeted motorcyclist using YOLO,” *ICT Express*, vol. 7, no. 1, pp. 104–109, 2021.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS SUMATERA UTARA  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007  
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI  
NOMOR : 2744/UN5.2.14.D/SK/SPB/2024  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 11 Juli 2024 perihal permohonan ujian skripsi:  
Nama : YUSRIANTONI  
NIM : 171402145  
Program Studi : Sarjana (S-1) Teknologi Informasi  
Judul Skripsi : Deteksi Helm Plat Sepeda Motor Menggunakan YOLOv8 Untuk Pencatatan Pelanggar Lalu Lintas
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.  
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.  
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.  
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026

MEMUTUSKAN

- Menetapkan :  
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:  
Ketua : Dr. Marischa Elveny S.TI, M.Kom  
NIP: 199003272017062001  
Sekretaris : Rossy Nurhasanah S.Kom., M.Kom  
NIP: 198707012019032016  
Anggota Penguji : Dedy Arisandi ST., M.Kom.  
NIP: 197908312009121002  
Anggota Penguji : Mohammad Fadly Syah Putra, B.Sc., M.Sc.  
NIP: 198301292009121003  
Moderator : -  
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

- Tembusan :
- 1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
  - 2. Yang bersangkutan
  - 3. Arsip

Medan  
Ditandatangani secara elektronik oleh:  
Dekan



Maya Silvi Lydia  
NIP 197401272002122001