

**PENERAPAN ALGORITMA YOLOV8 DALAM DETEKSI  
PENGGUNAAN HELM DAN PLAT NOMOR POLISI  
KENDARAAN PADA SEPEDA MOTOR  
BERBASIS ANDROID**

**SKRIPSI**

**VANISSYA ARBASHIKA PUTRI  
201402103**



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

**PENERAPAN ALGORITMA YOLOV8 DALAM DETEKSI  
 PENGGUNAAN HELM DAN PLAT NOMOR POLISI  
 KENDARAAN PADA SEPEDA MOTOR  
 BERBASIS ANDROID**

**SKRIPSI**

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

**VANISSYA ARBASHIKA PUTRI  
 201402103**



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

## **PERSETUJUAN**

Judul : Penerapan Algoritma YOLOv8 Dalam Deteksi Penggunaan Helm dan Plat Nomor Polisi Kendaraan pada Sepeda Motor Berbasis Aandroid

Kategori : Skripsi

Nama : Vanissya Arbashika Putri

Nomor Induk Mahasiswa : 201402103

Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi  
Universitas Sumatera Utara

Medan, 11 Juli 2024

Komisi Pembimbing:

Pembimbing 2

Pembimbing 1

Dr. Sawaluddin M.IT  
NIP. 195912311998021001

Dr. Marischa Elveny S.TI., M.Kom.  
NIP. 199003272017062001

Diketahui/disetujui oleh  
Program Studi S1 Teknologi Informasi  
Ketua,

Dedy Arisandi S.T., M.Kom.  
NIP. 197908312009121002

**PERNYATAAN**

PENERAPAN ALGORITMA YOLOV8 DALAM DETEKSI PENGGUNAAN  
HELM DAN PLAT NOMOR POLISI KENDARAAN PADA SEPEDA  
MOTOR BERBASIS ANDROID

**SKRIPSI**

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

Medan, 11 Juli 2024

Vanissya Ardashika Putri  
201402103

## ABSTRAK

Angka kecelakaan dan cedera pada pengendara sepeda motor saat ini tercatat meningkat sebesar 6,8% dari tahun 2022. Badan Pusat Statistik (BPS) Indonesia telah mendata bahwa sebagian besar kecelakaan fatal yang melibatkan pengendara sepeda motor disebabkan oleh penggunaan helm yang tidak memadai atau tidak digunakan sama sekali. Oleh karena itu, dibuatlah sistem yang dapat mendeteksi pelanggaran penggunaan helm secara real-time dan mencatat nomor plat kendaraan secara otomatis. Penilitian ini menggunakan metode You Only Look Once versi 8 (YOLOv8) dan memanfaatkan EasyOCR. YOLOv8 mampu mendeteksi objek helm dan plat nomor polisi kendaraan pada sepeda motor. EasyOCR digunakan untuk ekstraksi plat nomor polisi kendaraan. Hasil pengujian Model algoritma You Only Look Once versi 8 memperoleh nilai akurasi sebesar 96% dengan nilai precision rata-rata 96.5%, recall 100%, f1-score 98% pada deteksi penggunaan helm pada pengendara. Pada deteksi plat nomor polisi kendaraan memperoleh nilai akurasi sebesar 89.5% dengan nilai rata-rata precision 98%, recall 91%, dan f1-score 94%. Dengan metrik evaluasi yang diperoleh maka hasil sistem deteksi pada identifikasi pengguna helm dan plat nomor polisi kendaraan pada sepeda motor berkerja dengan sangat baik.

Kata Kunci: Helm, Plat Nomor Polisi, YOLOv8, Real-Time

*IMPLEMENTATION OF YOLOV8 ALGORITHM IN DETECTING HELMET  
USE AND VEHICLE LICENSE PLATES ON MOTORCYCLES  
BASED ON ANDROID*

**ABSTRACT**

*The number of motorcycle accidents and injuries has increased by 6.8% compared to 2022. The Central Bureau of Statistics (BPS) Indonesia has recorded that most fatal accidents involving motorcycle riders are due to inadequate helmet use or no helmet use at all. Therefore, a system has been developed to detect helmet violations in real time and automatically record vehicle license plate numbers. This research employs the You Only Look Once version 8 (YOLOv8) method and utilizes EasyOCR. YOLOv8 is capable of detecting helmet objects and vehicle license plates on motorcycles. EasyOCR is used for extracting vehicle license plate numbers. The testing results of the You Only Look Once version 8 algorithm model achieved an accuracy rate of 96% with an average precision of 96.5%, recall of 100%, and an F1-score of 98% in detecting helmet use among riders. In detecting vehicle license plate numbers, it achieved an accuracy rate of 89.5% with an average precision of 98%, recall of 91%, and an F1-score of 94%. With the obtained evaluation metrics, the detection system for identifying helmet users and vehicle license plates on motorcycles performs exceptionally well.*

*Keywords:* Helmet, License Plate, YOLOv8, Real-Time

## **UCAPAN TERIMA KASIH**

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas karunia, rahmat, dan kekuatan yang telah diberikan kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini dengan judul "Penerapan Algoritma YOLOv8 Dalam Deteksi Penggunaan Helm dan ", sebagai salah satu syarat untuk menyelesaikan Program Sarjana (S1) Jurusan Teknologi Informasi. Penulis menyadari bahwa skripsi ini tidak mungkin terselesaikan tanpa adanya dukungan, bantuan, bimbingan, dan nasehat dari berbagai pihak selama penyusunan skripsi ini. Pada kesempatan ini penulis menyampaikan terima kasih setulus-tulusnya kepada:

1. Orang tua penulis, Mama Dina Armayani, S.T., Alm. Papa Ir. Bambang Eka Basuki dan Ayah Ardian Nasution S.T. yang selalu memberikan dukungan moral dan material, serta doa yang tiada henti-hentinya.
2. Bapak Dedy Arisandi, S.T., M.Kom., selaku Ketua Program Studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Marischa Elveny S.TI., M.Kom. selaku dosen pembimbing 1 dan Bapak Dr. Sawaluddin M.IT selaku dosen pembimbing 2 yang telah memberikan bimbingan, arahan, dan masukan yang sangat berharga sepanjang proses penyusunan skripsi ini.
4. Bapak Prof. Dr. Drs. Opim Salim Sitompul, M.Sc. dan Bapak Niskarto Zendrato S.Kom., M.Kom selaku Dosen Pembanding yang telah memberikan kritik dan saran kepada penulis untuk penyempurnaan penulisan tugas akhir ini.
5. Bapak dan Ibu Dosen di lingkungan Program Studi S1 Teknologi Informasi, atas ilmu dan wawasan yang diberikan selama perkuliahan. Semua itu menjadi bekal yang sangat berharga dalam menyelesaikan skripsi ini.
6. Seluruh staff Fakultas Ilmu Komputer dan Teknologi Informasi yang telah membantu segala urusan administrasi selama perkuliahan.
7. Kakek Ahmad Sofyan dan Almh. Nenek Aryetti yang telah merawat, menjaga, dan memotivasi penulis agar menyelesaikan cepat dalam menyelesaikan penelitian ini.
8. Adik-adik penulis, yaitu Zeezee Auliazura Nasution dan Qhanza Aleasyifa Nasution yang telah menghibur, menemani, dan mendoakan penulis.

9. Om Yougi Akbar, S.Tr.Ak dan Tante Chairunnisa yang memberikan bimbingan, arahan, dan bantuan material.
10. Om Hery Hendra Subakti, ST dan Tante Lisma Yati Ningsih, Am. Kom yang telah menghibur, mendukung, dan mendoakan penulis.
11. Aulia Rahman Partomuan Sihite S.Kom yang telah menemani, mendengarkan keluh kesah, dan membantu penelitian penulis.
12. Ayu Kusmawati yang telah menjadi 911 penulis, mendengarkan keluh kesah, *life update*, mendukung, memberikan semangat dan mendoakan selama masa perkuliahan penulis.
13. Teman-teman grup capek pc, yaitu Ullayya Zhafirah, Tsabitah Muflizha, Nayla Rahmi Nasution, Ivan Tandella, Muhammad Iqbal Manalu, Muhammad Hatta Abdillah, dan Wahyu Sony Pratama, yang selalu memberikan semangat, bantuan, dan kebersamaan selama masa studi hingga penyusunan skripsi ini. Kalian adalah bagian penting dari perjalanan akademik penulis.
14. Teman-teman penulis, yaitu Ridha Arrahmi, Fildzah Zata Amani, Mhd Afifan Aly Rahman Saragih, Teruna Tegar Matondang dan M D Arbani Asfi Dalimunthe.
15. Abang dan kakak senior penulis, terutama Abangda Daniel Situmeang dan Abangda Geylfedra Matthew yang telah membantu dan memberikan saran untuk penelitian penulis.

Akhir kata, semoga skripsi ini dapat memberikan manfaat dan kontribusi positif bagi ilmu pengetahuan serta menjadi awal dari langkah-langkah saya selanjutnya dalam berkarya. Terima kasih.

Hormat saya,

Vanissya Ardashika Putri

## DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN	iv
ABSTRAK	v
ABSTRACT	vi
UCAPAN TERIMA KASIH	vii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.7. Metodologi Penelitian	5
BAB 2 LANDASAN TEORI	7
2.1. Helm Pengendara Sepeda Motor	7
2.2. Plat Nomor Polisi Kendaraan	7
2.3. Peraturan Keselamatan Berkendara	8
2.4. Computer Vision	8
2.5. Object Detection	8
2.6. Convolutional Neural Network	9
2.7. You Only Look Once	10
2.8. You Only Look Once versi 5	11

2.9.	You Only Look Once versi 8	11
2.10.	Optical Character Recognition	14
2.11.	<i>Confusion Matrix</i>	16
2.11.1.	<i>Accuracy</i>	17
2.11.2.	<i>Precision</i>	17
2.11.3.	<i>Recall</i>	17
2.11.4.	<i>F1-Score</i>	17
2.11.5.	<i>Mean Average Precision</i> (mAP)	17
2.12.	Penelitian Terdahulu	18
2.13.	Perbedaan Penelitian	20
BAB 3 ANALISIS DAN PERANCANGAN SISTEM		22
3.1.	Dataset	22
3.2.	Analisis Sistem	23
3.3.	<i>Image Acquisition</i>	24
3.3.1.	<i>Motorcyclist Detection</i>	24
3.3.2.	<i>Plate Detection</i>	24
3.4.	<i>Preprocessing</i>	25
3.4.1.	<i>Motorcyclist Detection</i>	25
3.4.2.	<i>Plate Detection</i>	30
3.5.	<i>Detection</i>	32
3.5.1.	<i>Package Installation</i>	32
3.5.2.	<i>Data Configuration</i>	33
3.5.3.	<i>Train Model</i>	33
3.6.	<i>Learned Model</i>	35
3.7.	<i>Text Extraction</i>	35
3.8.	<i>Deployment</i>	37
3.9.	<i>Output</i>	37

3.10. <i>System Interface</i>	37
BAB 4 IMPLEMENTASI DAN PENGUJIAN	40
4.1. Implementasi Sistem	40
4.2. Implementasi Data	41
4.3. Implementasi Model	42
4.3.1. Sistem Deteksi Pengendara Sepeda Motor	42
4.3.2. Sistem Deteksi Plat Nomor Polisi	43
4.4. Implementasi Antarmuka Aplikasi	44
4.4.1. Antarmuka Halaman <i>Home</i>	44
4.4.2. Antarmuka Halaman <i>Scan</i>	45
4.4.3. Antarmuka Halaman <i>Data</i>	45
4.5. Pengujian Sistem	46
4.5.1. Sistem Deteksi Pengendara Sepeda Motor	46
4.5.2. Sistem Deteksi Pengendara Sepeda Motor	55
BAB 5 KESIMPULAN DAN SARAN	64
5.1. Kesimpulan	64
5.2. Saran	64
DAFTAR PUSTAKA	66

**DAFTAR TABEL**

Tabel 2. 1 <i>Confusion Matrix</i>	17
Tabel 2. 2 Penelitian Terdahulu	19
Tabel 4. 1 Hasil Percobaan 150 <i>epoch</i>	42
Tabel 4. 2 Hasil Pengujian Sistem <i>Motorcyclist Detection</i>	46
Tabel 4. 3 <i>Confusion Matrix Helm Detection</i>	53
Tabel 4. 4 Hasil Pengujian Sistem <i>Plate Detection</i>	55
Tabel 4. 5 <i>Confusion Matrix Helm Detection</i>	62

## DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Convolution Neural Network (CNN)	9
Gambar 2. 2 Arsitektur You Only Look Once (YOLO)	10
Gambar 2. 3 Arsitektur YOLOv8	12
Gambar 2. 4 Perbandingan Algoritma You Only Look Once	13
Gambar 2. 5 Cara Kerja OCR	15
Gambar 2. 6 Arsitektur EasyOCR	16
Gambar 3. 1 Visualisasi Citra Pengegendaran	22
Gambar 3. 2 Contoh Citra Motor Untuk Deteksi Plat Nomor Polisi Kendaraan	23
Gambar 3. 3 Arsitektur Umum	24
Gambar 3. 4 Membuat <i>Bounding Box</i>	25
Gambar 3. 5 Memilih <i>Class</i>	26
Gambar 3. 6 Isi File.txt	26
Gambar 3. 7 Proses <i>Resizing</i>	27
Gambar 3. 8 Proses <i>Grayscale</i>	28
Gambar 3. 9 Proses <i>Flip Horizontal</i>	29
Gambar 3. 10 Proses <i>Blur</i>	30
Gambar 3. 11 Membuat <i>Bounding Box</i>	31
Gambar 3. 12 Memilih <i>Class</i>	31
Gambar 3. 13 Proses <i>Resizing</i>	32
Gambar 3. 14 Proses <i>Resizing</i>	35
Gambar 3. 15 Proses <i>Grayscale</i>	36
Gambar 3. 16 Proses <i>Thresholding</i>	37
Gambar 3. 17 Rancangan Halaman <i>Home</i>	38
Gambar 3. 18 Rancangan Halaman <i>Scan</i>	38
Gambar 3. 19 Rancangan Halaman Data	39
Gambar 4. 1 Contoh Data Citra Pengendara Motor	41
Gambar 4. 2 Grafik Hasil Percobaan <i>Batch Size 16</i>	43
Gambar 4. 3 Grafik Hasil <i>Batch Size 16</i>	44
Gambar 4. 4 Antarmuka Halaman <i>Home</i>	44

Gambar 4. 5 Antarmuka Halaman <i>Scan</i>	45
Gambar 4. 6 Antarmuka Halaman <i>Data</i>	46

## **BAB 1**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Angka kecelakaan dan cedera pada pengendara sepeda motor saat ini tercatat meningkat sebesar 6,8% dari tahun 2022. Berdasarkan data Komite Nasional Keselamatan Transportasi (KNKT) menyebutkan bahwa kasus kecelakaan yang melibatkan pengendara sepeda motor mencapai 198.251 jiwa pada tahun 2023. Meningkatnya angka tersebut menyebabkan penggunaan helm menjadi isu yang krusial dalam rangka menekan angka kecelakaan dan mencegah cedera serius pada pengendara. Undang-undang lalu lintas yang mengharuskan pengendara sepeda motor menggunakan helm telah diimplementasikan secara ketat di berbagai negara sebagai upaya untuk meningkatkan keselamatan pengendara. Kewajiban penggunaan helm di Indonesia telah diatur dalam Undang-Undang No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan. Namun, masih terdapat pelanggaran yang terjadi secara terus-menerus terkait dengan ketidakpatuhan pengendara dalam menggunakan helm, yang mengakibatkan dampak negatif bagi keselamatan masyarakat.

Badan Pusat Statistik (BPS) Indonesia telah mendata bahwa sebagian besar kecelakaan fatal yang melibatkan pengendara sepeda motor disebabkan oleh penggunaan helm yang tidak memadai atau tidak digunakan sama sekali. Dalam beberapa kota besar, upaya penegakan hukum dilakukan melalui kamera pengawas lalu lintas yang terpasang di sejumlah persimpangan jalan, memungkinkan pihak berwenang untuk mengidentifikasi pelanggaran dengan lebih efektif seperti memantau situasi lalu lintas dan melacak pelanggaran lalu lintas yang terjadi. Namun, sistem yang ada saat ini belum mampu mengidentifikasi pelanggaran penggunaan helm secara otomatis, membutuhkan campur tangan manusia untuk mengidentifikasi pelanggaran tersebut.

Sebagai solusi dari keterbatasan yang dihadapi oleh sistem yang digunakan saat ini, diperlukan metode otomatis untuk mendeteksi pelanggaran penggunaan helm pada pengendara sepeda motor. Dengan mengimplementasikan metode yang dapat memproses data secara *real-time*, petugas keamanan lalu lintas akan lebih mudah dalam mengidentifikasi dan merekam pelanggaran penggunaan helm tanpa harus melakukan input data secara manual. Hal ini akan membantu meningkatkan efisiensi penegakan hukum dan menurunkan tingkat pelanggaran secara signifikan.

Metode *image processing* sudah ada dilakukan dalam mendeteksi pelanggaran penggunaan helm pada pengendara sepeda motor melalui gambar atau rekaman video. Dalam hal ini, metode "You Only Look Once" (YOLO) menjadi hal baru dalam pendekalian objek melalui gambar dan video. Berbeda dengan metode lainnya, algoritma YOLO dapat mendeteksi objek dalam satu kali proses pengolahan sehingga lebih cepat dalam mendeteksi objek yang bergerak secara *real-time*.

Terdapat beberapa penelitian sebelumnya yang berfokus pada pendekalian penggunaan helm. Penelitian yang dilakukan oleh Zhang *et al.* (2019) menunjukkan bahwa penggunaan YOLO dalam mendeteksi pelanggaran penggunaan helm pada pengendara sepeda motor mendapatkan akurasi yang cukup tinggi yaitu mencapai 85%. Pengaplikasian penelitian ini menggunakan data rekaman video lalu lintas di beberapa persimpangan jalan utama.

Selain itu, penelitian yang dilakukan oleh Liu *et al.* (2020) menggunakan YOLO dalam mengenali pelanggaran penggunaan helm. Hasil penelitiannya menunjukkan bahwa metode YOLO mampu mengurangi waktu pemrosesan data secara signifikan dibandingkan dengan metode lain, seperti Support Vector Machine (SVM) dan Convolutional Neural Network (CNN). Hal ini berdampak pada peningkatan kinerja petugas keamanan lalu lintas dalam menindak pelanggaran dengan lebih efisien.

Pada tahun 2021 Wang *et al.* meneliti metode YOLO dalam menghadapi berbagai kondisi pencahayaan dan latar belakang gambar yang kompleks. Serangkaian uji coba telah dilakukan di berbagai kondisi jalan raya, termasuk pada malam hari dan cuaca buruk. Hasil penelitian tersebut menemukan bahwa YOLO mampu tetap mengenali pelanggaran penggunaan helm secara akurat tanpa terpengaruh oleh variasi kondisi lingkungan.

YOLO versi 8 merupakan pengembangan terbaru dari model YOLO yang menghadirkan sejumlah peningkatan performa dan akurasi deteksi. Dibandingkan

dengan versi sebelumnya, YOLO versi 8 menjanjikan pengoptimalan kecepatan dan akurasi deteksi yang lebih baik, memungkinkan identifikasi objek dengan lebih cepat dan presisi yang lebih tinggi. Dengan meningkatnya akurasi dan kecepatan deteksi, YOLO versi 8 memungkinkan sistem untuk mengidentifikasi pelanggaran penggunaan helm dengan lebih akurat dan efisien, memberikan solusi yang lebih handal dalam menangani isu keselamatan lalu lintas yang berkaitan dengan penggunaan helm pada pengendara sepeda motor.

Berdasarkan latar belakang tersebut, maka penulis mengangkat penelitian dengan judul **Penerapan Algoritma YOLOv8 dalam Deteksi Penggunaan Helm dan Plat Nomor Polisi Kendaraan pada Sepeda Motor**. Penelitian ini akan menganalisis performa algoritma YOLO versi 8 dalam mendeteksi penggunaan helm dan membaca plat nomor pada sepeda motor saat bergerak.

### **1.2. Rumusan Masalah**

Meningkatnya kasus kecelakaan pengendara bermotor pada tahun 2023 yang menyebabkan kecelakaan fatal dikarenakan oleh penggunaan helm yang tidak memadai atau tidak digunakan sama sekali. Ditengah banyaknya kasus yang telah terjadi masih banyak pengendara motor yang tidak menggunakan helm sehingga diperlukannya tindakan untuk mengurangi masalah tersebut. Saat ini pihak berwenang mencatat pelanggar secara manual dengan cara menilang pengemudi sepeda motor. Oleh karena itu, diperlukan sistem yang dapat mendeteksi pelanggaran penggunaan helm secara real-time dan mencatat nomor plat kendaraan secara otomatis.

### **1.3. Tujuan Penelitian**

Tujuan penelitian ini adalah untuk mengimplementasikan metode You Only Look Once (YOLO) versi 8 untuk deteksi penggunaan helm pada pengendara sepeda motor dan plat nomor polisi.

### **1.4. Batasan Masalah**

Berdasarkan permasalahan yang sudah disebutkan, peneliti membuat beberapa batasan masalah agar penelitian lebih terarah dan menghindari penyimpangan serta meluasnya lingkup permasalahan, yaitu:

- 1) Data yang digunakan merupakan citra pengguna kendaraan sepeda motor.
- 2) Penelitian ini akan mendeteksi pelanggaran pada penggunaan helm saat berkendara.
- 3) Pengambilan citra diambil dari lalu lintas di Kota Medan.
- 4) *Output* yang dihasilkan dari penelitian ini adalah data pelanggaran penggunaan helm dan plat nomor polisi kendaraan.

### **1.5. Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

- 1) Hasil penelitian ini diharapkan dapat membantu pihak yang berwenang sebagai penanggung jawab penegakan hukum lalu lintas dalam menindak pelanggaran lalu lintas.
- 2) Hasil penelitian ini diharapkan dapat dijadikan referensi dan sumber awal untuk penelitian selanjutnya.

### **1.6. Metodologi Penelitian**

Metodologi penelitian yang akan dilakukan pada penelitian ini yaitu:

#### **1) Studi Literatur**

Pada tahap ini, studi literatur dilakukan dengan mengumpulkan referensi dari jurnal, artikel, website dan sumber lainnya tentang identifikasi gambar, pengolahan gambar, You Only Look Once versi 8 dan Easy Optical Character Recognition.

#### **2) Analisis Permasalahan**

Pada tahap ini dilakukan analisis terhadap tahapan yang telah dilakukan pada studi literatur agar dapat memahami metode yang akan diimplementasikan dalam penelitian untuk membangun mesin deteksi penggunaan helm dan membaca plat nomor kendaraan sepeda motor.

#### **3) Perancangan Sistem**

Pada tahap ini dilakukan proses perancangan sistem berdasarkan analisis permasalahan yang telah dilakukan pada analisis permasalahan. Perancangan sistem meliputi perancangan arsitektur, pengumpulan data dan pembagian data training, data validation dan data testing.

**4) Implementasi**

Pada tahap ini dilakukan implementasi ke dalam bentuk source code berdasarkan analisis dan rancangan sistem untuk menghasilkan sistem mesin deteksi penggunaan helm, deteksi plat kemudian ekstraksi teks nomor kendaraan sepeda motor.

**5) Pengujian Sistem**

Pada tahap ini sistem yang telah diimplementasikan untuk mengetahui tingkat akurasi yang didapat menggunakan YOLOv8 dan EasyOCR dalam pengembangan deteksi helm dan plat nomor kendaraan.

**6) Dokumentasi dan Penyusunan Laporan**

Pada tahap ini dilakukan dokumentasi yaitu melampirkan laporan berupa video pengujian program dan penulisan laporan yang menjelaskan hasil dari penelitian yang telah dilakukan.

## **1.7. Metodologi Penelitian**

Sistem penulisan skripsi ini terdiri dari beberapa bagian yaitu:

### **BAB 1: PENDAHULUAN**

Pada bab ini berisikan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, serta metodologi dan sistematika penulisan.

### **BAB 2: LANDASAN TEORI**

Pada bab ini berisikan teori yang berhubungan dengan helm pengendara, plat nomor polisi, pelanggaran lalu lintas, *computer vision*, *object detection*, Convolutional Neural Network, You Only Look Once sebagai metode yang diimplementasikan dalam penelitian.

### **BAB 3: ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini berisikan penjelasan mengenai arsitektur umum yang digunakan pada penelitian ini dimulai dari tahap *data acquisition*, *pre-processing*, *training*, *testing*, *output*, sampai perancangan antarmuka sistem.

### **BAB 4: IMPLEMENTASI DAN PENGUJIAN**

Pada bab ini berisikan proses implementasi dan pengujian dari hasil rancangan sistem yang telah dijelaskan pada bab sebelumnya agar dapat memperlihatkan hasil penelitian ini dari pengujian apakah sudah mencapai tujuan.

**BAB 5: KESIMPULAN DAN SARAN**

Pada bab ini berisikan kesimpulan dari hasil penelitian yang telah dilakukan beserta saran yang dapat diterapkan pada penelitian kedepannya.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Helm Pengendara Sepeda Motor**

Helm merupakan perlengkapan keselamatan penting bagi pengendara sepeda motor, dirancang untuk melindungi kepala terhadap resiko benturan. Helm menjadi kewajiban bagi pengendara dengan ditetapkan didalam undang-undang oleh banyak negara. Indonesia mewajibkan pengendara sepeda motor harus memakai helm, untuk menjaga kepala dari benturan keras saat terjadi kecelakaan. Berikut merupakan dua jenis helm yang umum digunakan oleh pengendara sepeda motor yaitu *full face* dan *half face*.

##### *2.1.1. Helm Full Face*

*Helm full face* adalah jenis helm sepeda motor jenis helm sepeda motor yang dirancang untuk memberikan perlindungan maksimal kepada pengendara dengan menutupi seluruh bagian kepala, termasuk dagu dan wajah. Helm ini biasanya dilengkapi dengan pelindung wajah yang bisa dibuka dan ditutup, serta bantalan pelindung di bagian dalam untuk kenyamanan dan keamanan.

##### *2.1.2. Helm Half Face*

*Helm half face* juga dikenal sebagai helm open face, adalah jenis helm sepeda motor yang menutupi bagian atas dan sisi kepala, namun tidak menutupi wajah dan dagu secara penuh. Helm ini memberikan perlindungan dasar untuk kepala, tetapi tidak melindungi bagian bawah wajah, termasuk dagu dan mulut.

#### **2.2. Plat Nomor Kendaraan**

Plat nomor merupakan sebuah bagian yang disematkan pada setiap kendaraan bermotor sebagai identitas kendaraan. Plat nomor kendaraan ditetapkan penggunaannya di dalam undang-undang yang didalamnya terdapat kombinasi angka dan huruf dengan penyusunannya juga ditetapkan di dalam undang-undang. Plat nomor kendaraan

digunakan pihak kepolisian dalam melakukan identifikasi dan menyimpan catatan pelanggaran kendaraan.

### **2.3. Peraturan Keselamatan Berkendara**

Pada setiap negara hukum terdapat peraturan yang mengatur kegiatan lalu lintas sehingga tercapai keselamatan dan kenyamanan dalam berkendara. Di Indonesia peraturan terkait kegiatan berkendara secara spesifik mengatur terkait penggunaan helm pada pengendara sepeda motor. Undang-undang yang berkaitan tersebut diatur pada pasal 57 ayat (1) jo ayat (2) UU NO.22 Tahun 2009, selain itu pada undang-undang yang sama pasal 291 diatur hukuman yang diterapkan terhadap pelanggaran.

### **2.4. Computer Vision**

*Computer vision* adalah penggabungan dari *image processing* dan *pattern recognition* (Cosido et al., 2014), dimana hasil dari *computer vision* ini adalah membuat komputer dapat memahami sebuah citra. Sebagian besar tugas dari *Computer Vision* proses untuk mendapatkan informasi dari sebuah kejadian maupun sebuah deskripsi dari sebuah kejadian (Wiley & Lucas, 2018).

Menurut Aloimonos & Rosenfeld (1991) *Computer vision* adalah proses untuk menggali dan mendapatkan informasi tentang beberapa kejadian dengan menganalisa gambar dari kejadian tersebut. Untuk melakukan proses pada gambar dengan (digital) komputer, harus dengan mengubah gambar tersebut menjadi sebuah gambar digital yang dimana merupakan kumpulan dari angka angka yang mewakili nilai kecerahan atau warna pada titik - titik dalam sebuah gambar. Didalam *computer vision*, gambar adalah sebuah data awal untuk mendeteksi kejadian sedangkan *output* dari *computer vision* adalah sebuah informasi berupa kejadian yang diberikan.

### **2.5. Object Detection**

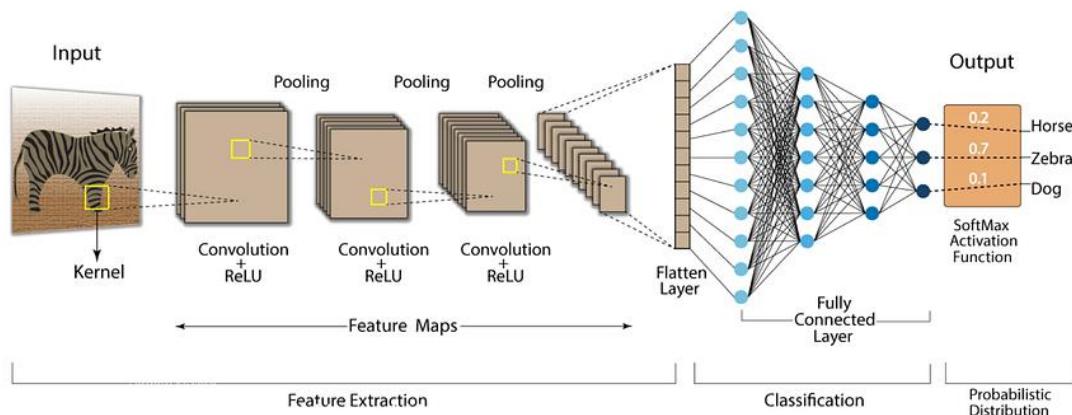
*Object detection* merupakan salah satu dari cabang computer vision yang bertugas mendeteksi objek visual dari kelas tertentu dalam citra digital. *Object detection* bertujuan untuk mengembangkan komputasi model dan menyediakan Teknik pengetahuan yang dibutuhkan oleh aplikasi computer vision (Zou et al., 2023).

## 2.6. Convolutional Neural Network

Convolutional Neural Network adalah salah satu jenis dari metode deep learning yang mampu mengidentifikasi dan klasifikasi sebuah objek secara mandiri (Zhang & Zhu, 2018). CNN terdiri dari sejumlah besar lapisan tersembunyi yang masing-masing melakukan komputasi matematika pada input dan menghasilkan output yang dijaskan input pada lapisan berikutnya (Miranda et al., 2020).

Pada Gambar 2.1 memperlihatkan Arsitektur CNN yaitu dimana CNN menerima gambar input, seperti gambar zebra, dan menerapkan kernel (filter) pada area kecil gambar untuk mendeteksi fitur spesifik. Proses ini dimulai dengan lapisan konvolusi yang menghasilkan peta fitur, diikuti oleh fungsi aktivasi ReLU (Rectified Linear Unit) yang memperkenalkan non-linearitas. Lapisan pooling kemudian mengurangi dimensi peta fitur sambil mempertahankan fitur penting, dan proses konvolusi dan pooling ini diulang beberapa kali untuk membentuk lapisan peta fitur yang lebih dalam.

Setelah beberapa lapisan konvolusi dan pooling, peta fitur yang dihasilkan diratakan menjadi satu vektor panjang melalui lapisan flatten. Vektor ini kemudian melewati satu atau lebih lapisan fully connected (dense) untuk melakukan tugas klasifikasi berdasarkan fitur yang diekstraksi. Pada lapisan output, fungsi aktivasi softmax menghasilkan distribusi probabilitas atas kemungkinan kelas, seperti Kuda, Zebra, dan Anjing. Probabilitas output menunjukkan kemungkinan bahwa gambar input termasuk dalam setiap kelas, dan dalam contoh ini, model memprediksi gambar adalah Zebra dengan probabilitas tertinggi sebesar 0.7.

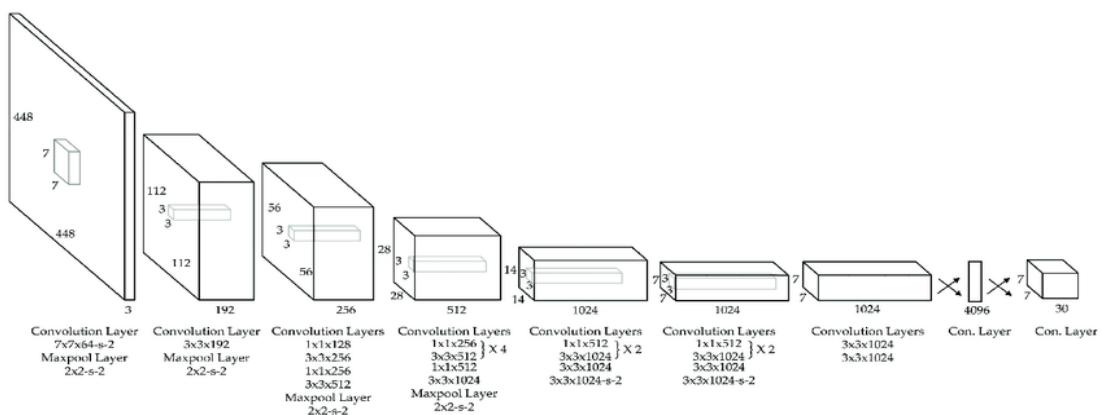


**Gambar 2. 1** Arsitektur Convolution Neural Network (CNN)

(Sumber: medium.com)

## 2.7. You Only Look Once

Algoritma You Only Look Once merupakan metode yang digunakan dalam pengolahan citra dan pengenalan objek dalam visi computer. YOLO pertama kali dikembangkan oleh (Redmon et al., 2015a) yang menyajikannya sebagai masalah regresi yang memisahkan *bounding box* dan *class probabilities* secara spasial. YOLO menggunakan jaringan konvolusi tunggal yang secara bersamaan memprediksi beberapa kotak pembatas dan probabilitas kelas untuk setiap kotak dari satu gambar dalam satu evaluasi sehingga mencapai kecepatan *real-time* dan akurasi yang tinggi. YOLO sangat cepat hingga mampu memproses gambar pada 45 *frame per second*.



**Gambar 2. 2** Arsitektur You Only Look Once (YOLO)  
(Sumber: Gnanaprakash et al., 2021)

Dapat dilihat pada Gambar 2.2, arsitektur dari model YOLO (You Only Look Once) untuk deteksi objek dalam gambar. Model ini mengambil gambar input berukuran 448x448 dengan 3 saluran warna (RGB). Tahap pertama adalah lapisan konvolusi dengan kernel 7x7 dan 64 filter yang menghasilkan keluaran berukuran 192x192x64, diikuti oleh lapisan max pooling. Model kemudian melanjutkan dengan beberapa lapisan konvolusi dan pooling secara berturut-turut, mengurangi dimensi spasial sambil meningkatkan dimensi kedalaman fitur, yang mencakup lapisan dengan kernel 3x3 dan filter yang bervariasi hingga 1024 pada lapisan terakhir. Proses ini menangkap berbagai fitur dari gambar input melalui deteksi tepi, tekstur, dan pola kompleks lainnya.

Setelah melalui serangkaian lapisan konvolusi dan pooling, hasilnya diratakan melalui lapisan fully connected dengan ukuran 4096. Lapisan terakhir dari model adalah lapisan konvolusi yang menghasilkan keluaran berukuran 7x7x30, yang mencakup prediksi *bounding box* dan probabilitas kelas untuk setiap kotak deteksi. Arsitektur ini

memungkinkan YOLO untuk melakukan deteksi objek secara efisien dengan satu forward pass melalui jaringan, memberikan keseimbangan antara kecepatan dan akurasi yang menjadikannya pilihan populer dalam tugas deteksi objek real-time.

### **2.8. You Only Look Once versi 5**

YOLOv5 dirilis oleh Glenn Jocher seorang CEO dan peneliti dari Ultralytics pada 9 juni 2020 berselang dua bulan dari perilisan versi sebelumnya. Berbeda dari versi sebelumnya perilisan versi ini tidak diterbitkan dengan tulisan melainkan hanya merilis implementasi sumber terbuka pada github. YOLO versi ini terdiri dari 4 versi utama yaitu nano (n), kecil (s), sedang (m) dan besar (l). Setiap versi tersebut memiliki waktu yang berbeda untuk pelatihan.

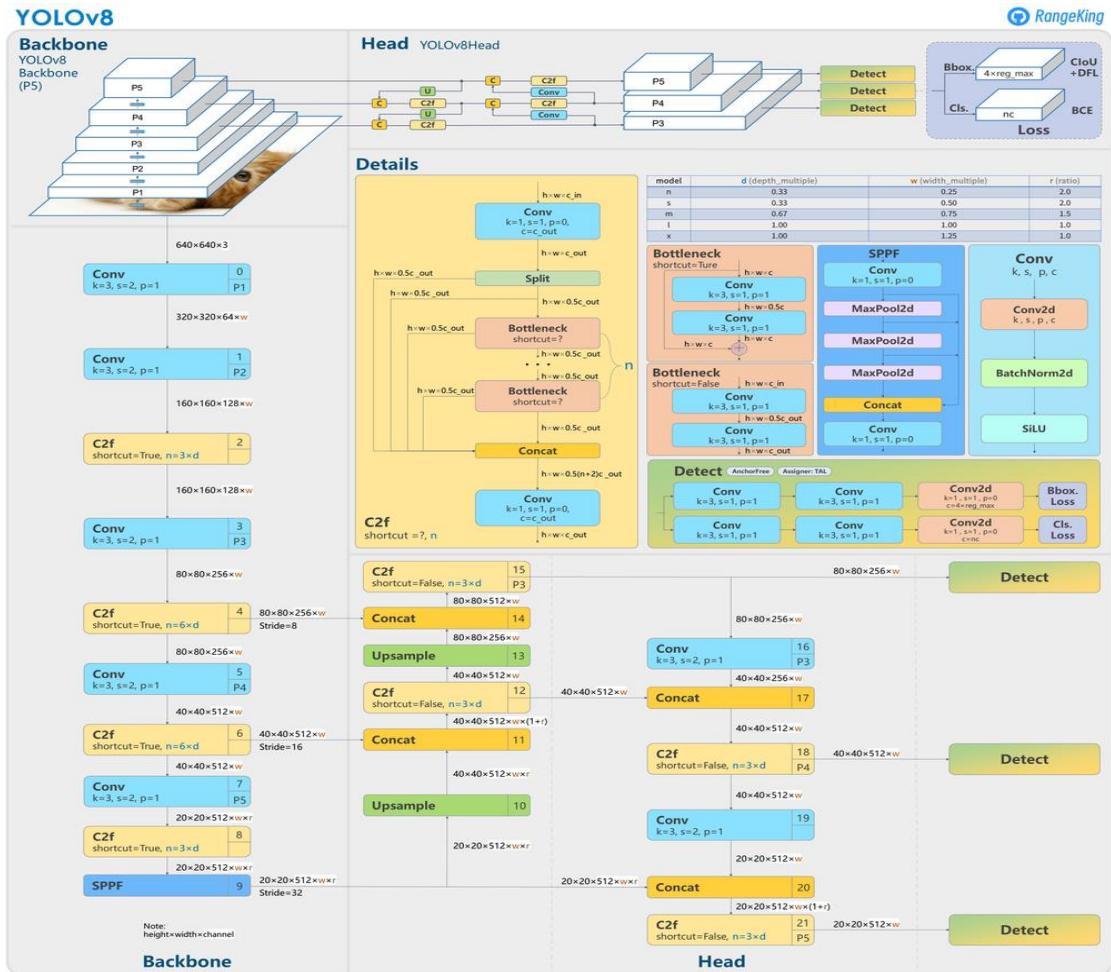
Perbedaan antara You Only Look Once versi kelima dengan versi sebelumnya terletak pada penggunaan Bahasa Python dan penggunaan kerangka kerja Pytorch. Perbedaan tersebut membuat YOLO versi ini lebih bebas untuk digunakan pada berbagai model. Pytorch memberikan hasil yang mempermudah pengembangan untuk memodifikasi arsitektur dan diterapkan secara langsung. YOLO versi kelima ini menggunakan konfigurasi file .yaml berbeda dengan versi sebelumnya yang menggunakan konfigurasi file .cfg.

### **2.9. You Only Look Once versi 8**

YOLOv8 dirilis pada Januari 2023 oleh Ultralytics, perusahaan yang mengembangkan YOLOv5. YOLOv8 menyediakan lima versi skala: YOLOv8n (nano), YOLOv8s (kecil), YOLOv8m (sedang), YOLOv8l (besar) dan YOLOv8x (ekstra besar) (Terven & Cordova-Esparza, 2024). YOLOv8 mendukung berbagai tugas penglihatan seperti deteksi objek, segmentasi, estimasi pose, pelacakan, dan klasifikasi. Gambar 4 menunjukkan detail arsitektur YOLOv8. YOLOv8 menggunakan tulang punggung yang mirip dengan YOLOv5 dengan beberapa perubahan pada CSPLayer, yang sekarang disebut modul C2f.

YOLOv8 menggunakan model bebas jangkar dengan kepala terpisah untuk memproses tugas objektivitas, klasifikasi, dan regresi secara independen. Desain ini memungkinkan setiap cabang untuk fokus pada tugasnya dan meningkatkan akurasi model secara keseluruhan. Di lapisan keluaran YOLOv8, mereka menggunakan fungsi sigmoid sebagai fungsi aktivasi untuk skor objektivitas, yang mewakili probabilitas

bahwa kotak pembatas berisi objek. Ia menggunakan fungsi softmax untuk probabilitas kelas, mewakili probabilitas objek yang termasuk dalam setiap kelas yang mungkin.



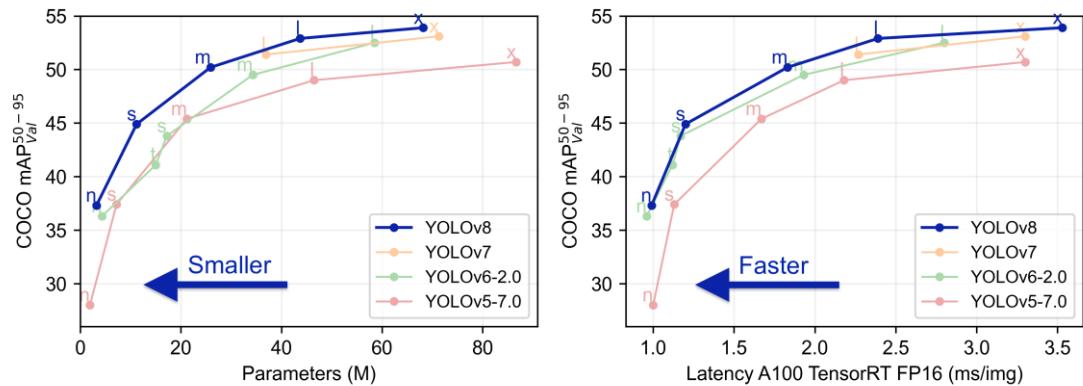
**Gambar 2.3** Arsitektur YOLOv8

(Sumber: roboflow.com)

Arsitektur YOLOv8 pada Gambar 2.3, yang terdiri dari dua bagian utama yaitu *backbone* dan *head*. Bagian *backbone* YOLOv8 terdiri dari beberapa lapisan konvolusi dan blok C2f (Cross Stage Partial Network), yang membantu dalam ekstraksi fitur dari gambar input. Backbone ini berfungsi untuk menangkap berbagai tingkat fitur dari gambar, mulai dari fitur yang sederhana hingga yang kompleks, dengan memprosesnya melalui beberapa tahap lapisan konvolusi dan blok C2f. Lapisan konvolusi pertama (Conv) dengan kernel 3x3 dan stride 2 mengurangi dimensi gambar, diikuti oleh beberapa lapisan konvolusi dan blok C2f untuk mengolah fitur lebih lanjut, dengan ukuran output yang terus mengecil tetapi kedalaman fitur yang semakin meningkat. Proses ini mencakup blok C2f dengan dan tanpa shortcut, yang meningkatkan kemampuan jaringan dalam menangani berbagai ukuran objek.

Pada bagian *head* YOLOv8 terdiri dari beberapa lapisan untuk deteksi objek. *Head* ini terdiri dari beberapa lapisan konvolusi dan blok C2f yang bertugas untuk memproses fitur yang telah diekstraksi oleh *backbone* dan membuat prediksi bounding box dan kelas objek. *Head* juga mencakup beberapa operasi seperti *concatenation* (Concat) dan upsampling untuk menggabungkan fitur dari berbagai lapisan dan memperbesar resolusi fitur untuk deteksi objek yang lebih akurat. Terdapat tiga lapisan deteksi yang menghasilkan prediksi akhir untuk *bounding box*, skor objek, dan kelas objek. Selain itu, arsitektur ini juga menggunakan mekanisme SPPF (Spatial Pyramid Pooling-Fast) yang menggabungkan informasi dari berbagai skala spasial untuk meningkatkan kemampuan deteksi objek pada berbagai ukuran. Diagram ini memberikan gambaran mendetail tentang setiap lapisan, parameter yang digunakan, dan bagaimana data diproses melalui jaringan untuk menghasilkan deteksi objek yang akurat.

YOLOv8 sangat cocok untuk berbagai tugas seperti deteksi dan pelacakan objek, segmentasi instance, klasifikasi gambar, dan estimasi pose karena dirancang dengan cepat, akurat, dan mudah digunakan.



**Gambar 2. 4** Perbandingan Algoritma You Only Look Once  
(Sumber: ultralytics.com)

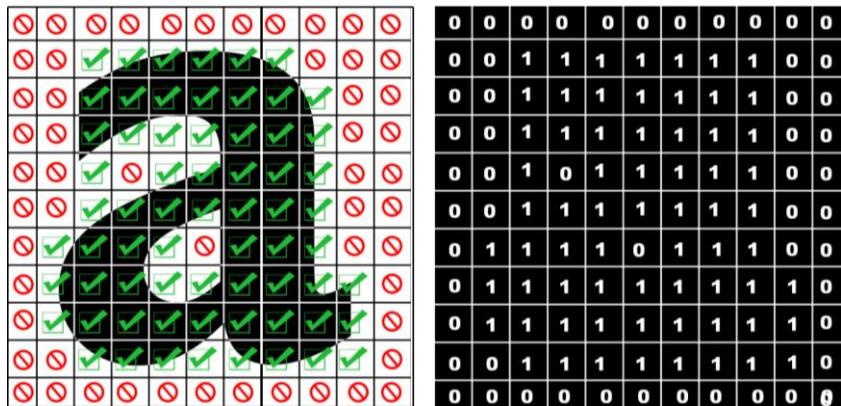
Gambar 2.4 menampilkan dua grafik yang membandingkan performa berbagai versi model YOLO (You Only Look Once) pada dataset COCO dalam hal akurasi (mAP) dan parameter serta latensi. Grafik pertama di sebelah kiri menunjukkan hubungan antara jumlah parameter dalam model (dalam juta) dan akurasi deteksi objek (COCO mAP<sub>50-95</sub> val). Grafik ini menunjukkan bahwa model YOLOv8 (ditandai dengan garis biru tebal) memiliki performa yang lebih baik dibandingkan model YOLOv7, YOLOv6-2.0, dan YOLOv5-7.0 dengan jumlah parameter yang lebih

sedikit. Sedangkan grafik kedua di sebelah kanan menunjukkan hubungan antara latensi (dalam milidetik per gambar) dan akurasi deteksi objek. Grafik ini mengindikasikan bahwa model YOLOv8 memiliki latensi yang lebih rendah (lebih cepat) dibandingkan model lainnya untuk mencapai akurasi yang sama atau lebih tinggi. Kesimpulannya, YOLOv8 menonjol dengan akurasi deteksi objek yang lebih tinggi, menggunakan lebih sedikit parameter, dan memiliki latensi yang lebih rendah dibandingkan dengan versi YOLO lainnya yang diuji.

## 2.10. Optical Character Recognition

Optical Character Recognition (OCR) adalah konversi elektronik dari gambar teks yang diketik, ditulis tangan, atau dicetak menjadi teks yang dikodekan mesin.(Karandish, 2022). Dengan OCR, sejumlah besar dokumen berbasis kertas, dalam berbagai bahasa dan format, dapat didigitalkan menjadi teks yang dapat dibaca mesin yang tidak hanya membuat penyimpanan lebih mudah tetapi juga membuat data yang sebelumnya tidak dapat diakses tersedia bagi siapa saja dengan sekali klik.

Cara kerja OCR yaitu dengan membagi gambar karakter teks menjadi beberapa bagian dan membedakan antara wilayah kosong dan tidak kosong (Anderson, 2021). Bergantung pada font atau skrip yang digunakan untuk surat tersebut, checksum dari matriks yang dihasilkan selanjutnya diberi label sesuai dengan karakter dalam gambar. Dapat dilihat pada Gambar 2.5 menunjukkan representasi digital dari huruf "a" menggunakan matriks biner dan grid piksel. Bagian kiri gambar menampilkan sebuah grid dengan tanda centang hijau dan larangan merah. Tanda centang hijau menunjukkan piksel yang aktif atau bernilai 1, yang membentuk huruf "a", sedangkan tanda larangan merah menunjukkan piksel yang tidak aktif atau bernilai 0, yang merupakan bagian dari latar belakang. Bagian kanan gambar menampilkan representasi biner dari grid di sebelah kiri, di mana nilai 1 menunjukkan piksel yang aktif dan nilai 0 menunjukkan piksel yang tidak aktif. Proses ini dimulai dengan digitisasi huruf "a" ke dalam bentuk grid piksel, kemudian dikonversi ke dalam format biner, dan divisualisasikan baik secara grafis maupun numerik. Gambar 2.5 menggambarkan dasar representasi gambar dalam komputer, di mana setiap gambar dipecah menjadi piksel dan disimpan dalam bentuk nilai biner untuk pemrosesan dan penyimpanan yang efisien.



**Gambar 2. 5 Cara Kerja OCR**

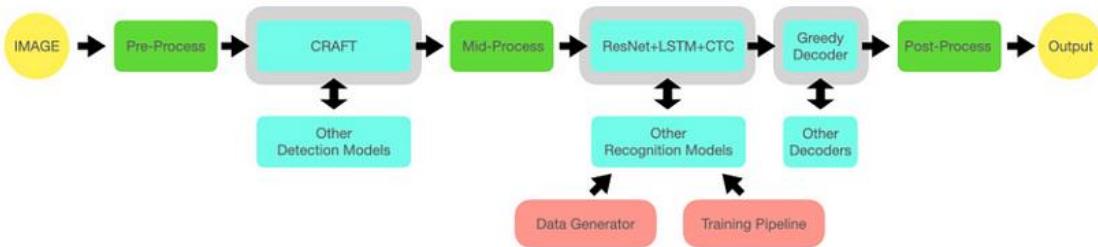
(Sumber: Anderson, 2021)

Easy Optical Character Recognition (EasyOCR) dibangun dengan PyTorch untuk mengendalikan *backend*-nya agar memungkinkan *deep learning*. EasyOCR dapat digunakan dengan berbagai bahasa, model deteksi dan identifikasi teks. EasyOCR terdiri dari 3 komponen utama yaitu *feature extraction* (Resnet), *sequence labeling* (LSTM), dan *decoding* (CTC) yang akan berfungsi sebagai satu unit agar memungkinkan EasyOCR mengekstrak teks dari gambar dengan efektif. Proses dimulai dengan mengambil gambar yang mengandung teks sebagai input. Pada tahap praproses, gambar dipersiapkan melalui langkah-langkah seperti normalisasi warna, resizing, dan penghapusan noise. Selanjutnya, model CRAFT (Character Region Awareness for Text detection) digunakan untuk mendeteksi area dalam gambar yang kemungkinan besar mengandung teks, menghasilkan peta region teks. Model deteksi teks lainnya juga dapat digunakan sebagai alternatif untuk meningkatkan akurasi.

Setelah itu, tahap mid-proses dilakukan untuk menggabungkan atau menyesuaikan region teks sebelum pengenalan karakter. Proses pengenalan teks melibatkan kombinasi dari tiga teknologi utama: ResNet (Residual Network) untuk ekstraksi fitur, LSTM (Long Short-Term Memory) untuk mengenali urutan karakter, dan CTC (Connectionist Temporal Classification) untuk mengelola pengenalan urutan karakter yang tidak tersegmen dengan jelas. Model pengenalan lain juga dapat digunakan untuk melengkapi atau meningkatkan akurasi.

Setelah pengenalan karakter, Greedy Decoder mengonversi output model pengenalan menjadi urutan karakter yang terbaca, meskipun decoder alternatif dapat digunakan untuk proses decoding yang lebih presisi. Tahap akhir adalah pasca-proses, di mana teks yang dikenali dibersihkan dan diformat sebelum dihasilkan sebagai output

akhir. Selain itu, arsitektur ini didukung oleh komponen seperti data generator yang menghasilkan data latih dan training pipeline yang melatih serta mengoptimalkan performa model deteksi dan pengenalan teks. Dengan gabungan komponen-komponen ini, EasyOCR mampu mendeteksi dan mengenali teks dalam gambar dengan efisien dan akurat. Untuk lebih lanjut dapat dilihat pada Gambar 2.6.



**Gambar 2.6** Arsitektur EasyOCR  
(Sumber: medium.com)

### 2.11. *Confusion Matrix*

*Confusion matrix* merupakan metode yang digunakan untuk mengevaluasi kinerja model atau algoritma yang digunakan dalam memprediksi kategori atau kelas dari data yang diberikan. *Confusion matrix* berisikan informasi mengenai jumlah prediksi yang benar dan salah dari model tersebut. Hubungan antara variabel aktual dan prediksi dapat dilihat lebih jelas pada Tabel 2.1, hasil prediksi digambarkan menjadi TP, TN, FP, dan FN dimana:

1. TP (*True Positive*) merupakan data yang bernilai positif dan menghasilkan prediksi positif.
2. TN (*True Negative*) merupakan data yang bernilai negatif dan menghasilkan prediksi juga negatif.
3. FP (*False Positive*) merupakan data yang bernilai negatif, tetapi menghasilkan prediksi positif.
4. FN (*False Negative*) merupakan data yang bernilai positif, tetapi menghasilkan prediksi negatif.

**Tabel 2. 1 Confusion Matrix**

<b>Prediksi \ Aktual</b>	<b>Positif</b>	<b>Negatif</b>
<b>Positif</b>	TP	FP
<b>Negatif</b>	FN	TN

#### 2.11.1. Accuracy

*Accuracy* adalah metrik yang menyajikan akurasi model memprediksi dengan tepat. *Accuracy* mengukur kemampuan model dengan membandingkan hasil prediksi yang tepat dengan seluruh data.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

#### 2.11.2. Precision

*Precision* adalah metrik yang menyajikan persentase prediksi positif yang benar. *Precision* yaitu kemampuan model untuk mengenali objek yang relevan.

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

#### 2.11.3. Recall

*Recall* adalah metrik persentase prediksi positif benar diantara semua *ground truth* yang ada. *Recall* merupakan kemampuan model untuk mengidentifikasi semua kasus yang relevan, atau semua *bounding box ground truth*.

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

#### 2.11.4. F1-Score

*F1-Score* merupakan gabungan antara *precision* dan *recall*.

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

#### 2.11.5. Mean Average Precision (mAP)

*Mean Average Precision* (mAP) merupakan metrik penting untuk mengevaluasi kinerja model deteksi objek. Metrik ini mengukur seberapa baik *bounding box* yang diprediksi cocok dengan *ground truth* di berbagai *class*. Jika nilai mAP yang tinggi maka kinerjanya baik dalam mendekripsi dan melokalisasi objek (Redmon et al., 2015). Dalam

You Only Look Once versi 8, metrik ini digunakan untuk mengukur prediksi model terhadap data *ground truth*. Dengan mengevaluasi mAP, peniliti dapat menilai secara kuantitatif kemampuan deteksi dari model yang digunakan.

## 2.12. Penelitian Terdahulu

Beberapa penelitian terdahulu mengenai pendekripsi helm dan plat nomor kendaraan bermotor sudah pernah dilakukan diantaranya adalah penelitian oleh Sankaraiah et al. (2023) yang membuat sistem deteksi pengendara tanpa helm menggunakan video sebagai input. Sistem tersebut mampu mengkstrak dan menampilkan nomor plat kendaraan jika pengendara sepeda motor dalam video tidak mengenakan helm. Metode deteksi objek dengan arsitektur YOLO v3 dan OCR untuk mengekstrak nomor plat.

Penelitian oleh Chellsya et al. (2023) menggunakan metode YOLO Tiny3 untuk mendekripsi pengendara motor yang menggunakan helm dan yang tidak menggunakan helm. Meskipun Yolo Tiny3 mencapai tingkat akurasi 100% pada tiga skenario pengujian dengan waktu proses rata-rata 0.2 detik, terdapat beberapa keterbatasan pada metode ini yaitu cenderung mengalami kesalahan ketika gambar buram, objek terlalu jauh, atau ketika dua objek berdekatan tidak dianggap dalam satu region helm. Hasil pengujian metode ini menunjukkan bahwa Yolo Tiny3 dapat memberikan tingkat presisi sebesar 88%.

Adapun penelitian oleh Khoiriyah et al. (2023) menunjukkan bahwa pendekripsi objek di area kepala menggunakan Convolutional Neural Network (CNN) memiliki tingkat akurasi rata-rata deteksi helm sebesar 89.04% dan avg\_loss sebesar 1.2%. Penelitian oleh Aprilino & Amin (2022) menggunakan metode YOLO v3 dalam mendekripsi objek plat nomor pada gambar kendaraan, dengan tingkat kepercayaan yang tinggi pada threshold 0.5 dan Tesseract menunjukkan ketergantungannya pada tahap pre-processing, khususnya proses grayscaling dan thresholding gambar sebelum melakukan OCR.

Dalam penelitian yang dilakukan oleh Illmawati & Hustinawati, (2022) menggunakan Algoritma YOLOv5 dan TesseractOCR untuk menemukan dan mengkategorikan plat nomor kendaraan ganjil-genap pada video lalu lintas pada siang hari dengan pencahayaan terbaik. Akurasi ekstraksi karakter plat nomor adalah 95,45 persen untuk teks terdeteksi dan 97,2 persen untuk keberhasilan program menentukan kategori plat nomor. Penelitian oleh Tun et al. (2020) ini membahas pendekatan hibrid

untuk mendeteksi penggunaan helm dan rompi keselamatan. YOLO digunakan untuk deteksi manusia, sementara SVM dengan HOG berbasis superpixel dan deskriptor warna digunakan untuk mendeteksi helm dan rompi.

**Tabel 2. 2** Penelitian Terdahulu

No.	Penulis	Judul	Tahun	Keterangan
1.	Sankaraiah et al.	<i>Artificial Intelligence Approach For Helmet Detection With Number Plate Extraction Using YOLO</i>	2023	Penelitian ini menggunakan metode YOLO yang digunakan untuk mengenali sepeda motor, orang, helm, dan nomor plat. Jika pengendara tidak mengenakan helm, OCR digunakan untuk mengekstrak nomor plat, dan tidak hanya karakternya, tetapi juga bingkai tempat ekstraksi dilakukan disimpan untuk penggunaan lain.
2.	Chellsya et al.	<i>Implementation Of Computer Vision In Detecting Violations Of Not Wearing Helmet On Motorcycles</i>	2023	Penelitian ini menggunakan algoritma Yolo Tiny3 untuk mendeteksi pengendara sepeda motor dan mencapai tingkat akurasi 100% pada tiga skenario pengujian dengan waktu proses rata-rata 0.2 detik, terdapat beberapa keterbatasan. Hasil pengujian menunjukkan bahwa Yolo Tiny3 dapat memberikan tingkat presisi sebesar 88%.
3.	Khoiriyah et al.	Deteksi Pengendara Motor Tanpa Menggunakan Helm Dengan Algoritma Deep Learning YOLO	2023	Penelitian ini menunjukkan bahwa pendekslsian objek di area kepala pengendara sepeda motor menggunakan metode Convolutional Neural Network mendapatkan tingkat akurasi rata-rata deteksi helm sebesar 89.04% dan avg_loss sebesar 1.2%.

**Tabel 2. 2** Penelitian Terdahulu

No.	Penulis	Judul	Tahun	Keterangan
4.	Aprilino & Amin	Implementasi Algoritma YOLO dan Tesseract OCR Pada Sistem Deteksi Plat Nomor Otomatis	2022	Penelitian ini menggunakan metode YOLOv3 dalam mendeteksi objek plat nomor pada gambar kendaraan, dengan tingkat kepercayaan yang tinggi pada threshold 0.5 dan Tesseract menunjukkan ketergantungannya pada tahap pre-processing, khususnya proses grayscaling dan thresholding gambar sebelum melakukan OCR.
5.	Illmawati & Hustinawati.	YOLO V5 for Vehicle Plate Detection in DKI Jakarta	2022	Dengan menggunakan algoritma YOLOv5 dan Tesseract OCR, penelitian ini menemukan dan mengkategorikan plat nomor kendaraan yang tidak teratur pada video lalu lintas. Akurasi ekstraksi karakter plat nomor adalah 95,45% untuk teks terdeteksi dan 97,2% untuk keberhasilan program menentukan kategori plat nomor.
6.	Tun et al.	Safety Helmet and Vest Wearing Detection Approach by Integratinig YOLO and SVM	2020	Penilitian ini membahas pendekatan hibrid untuk mendeteksi penggunaan helm dan rompi keselamatan. YOLO digunakan untuk deteksi manusia, sementara SVM dengan HOG berbasis superpixel dan deskriptor warna digunakan untuk mendeteksi helm dan rompi.

### 2.13. Perbedaan Penelitian

Penelitian ini memiliki perbedaan dengan penelitian sebelumnya yaitu, pada penelitian (Khoiriyah et al., 2023) menggunakan metode CNN untuk mendeteksi penggunaan helm dan (Illmawati & Hustinawati, 2022) menggunakan metode You Only Look Once

versi 5 untuk melakukan pengenalan plat nomor kendaraan, sedangkan pada penelitian ini menggunakan metode YOLO versi 8 untuk mendeteksi penggunaan helm dan plat nomor kendaraan pada sepeda motor.

Terdapat juga perbedaan pada ekstraksi teks yang digunakan, dimana pada penelitian (Aprilino & Amin, 2022) menggunakan Tesseract OCR, sedangkan pada penelitian ini menggunakan EasyOCR. Pada penelitian ini akan menghasilkan output berupa bukti pelanggaran dan hasil ekstraksi plat nomor polisi kendaraan sepeda motor.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

Pada bab 3 ini menjelaskan tentang analisis dan perangcangan sistem untuk melakukan deteksi helm dan plat kendaraan sepeda motor. Bab ini akan membahas mengenai sumber data yang digunakan, proses input, pra-pemrosesan data, pelatihan dan pengujian, serta hasil yang didapatkan.

#### **3.1. Dataset**

Pada penelitian ini menggunakan data yang diambil melalui pengumpulan manual pada lalu lintas Kota Medan. Dapat dilihat pada Gambar 3.1 pada gambar sebelah kiri terdapat pengendara sepeda motor yang tidak menggunakan helm dan tidak terlihat jelas plat nomor polisi kendaraannya. Pada gambar sebelah kanan merupakan citra pengendara motor yang menggunakan helm dan terlihat jelas plat nomor polisi kendaraannya.



**Gambar 3. 1** Visualisai Citra Pengegenda

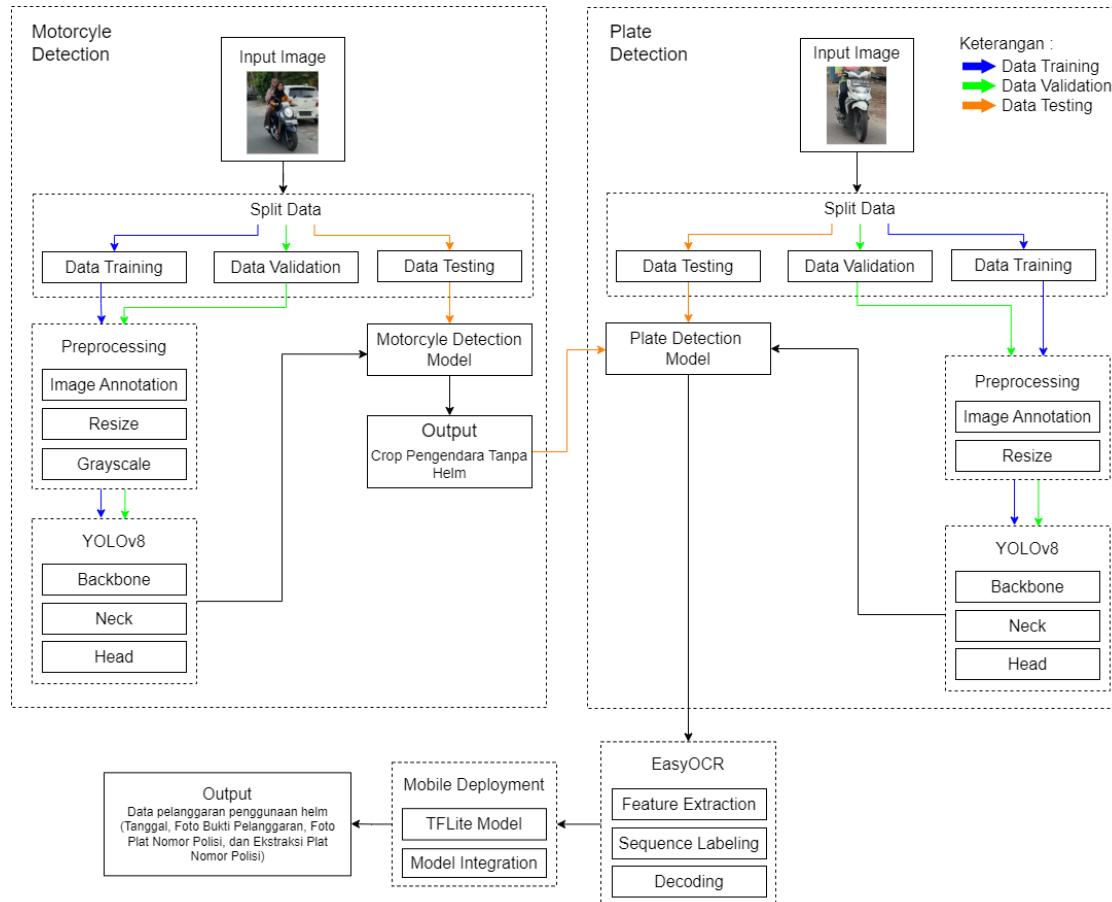
Dapat dilihat pada Gambar 3.2 terdapat contoh citra motor yang digunakan untuk *training* deteksi plat nomor kendaraan pada sepeda motor. Pada gambar sebelah kanan merupakan plat nomor polisi warna hitam dan pada gambar sebelah kiri merupakan plat nomor polisi berwarna putih.



**Gambar 3. 2** Contoh Citra Motor Untuk Deteksi Plat Nomor Polisi Kendaraan

### 3.2. Analisis Sistem

Pada penelitian ini dilakukan beberapa tahapan, Gambar 3.2 menunjukkan arsitektur umum penelitian ini. Data pengendaran sepeda motor dan plat nomor kendaraan sepeda motor akan dikumpulkan dalam bentuk gambar kemudian akan dibagi menjadi data train, data validation, dan data test.



**Gambar 3. 3** Arsitektur Umum

### 3.3. *Image Acqusition*

#### 3.3.1. *Motorcyclist Detection*

Tahap awal yang dilakukan adalah mengumpulkan data input yang akan digunakan sesuai dengan kebutuhan peneliti. Penilitian ini menggunakan citra gambar pengendara sepeda motor yang diambil dari lalu lintas Kota Medan. Kemudian data yang telah dikumpulkan dibagi menjadi tiga bagian, yaitu data *training*, data *validation* dan data *testing*.

#### 3.3.2. *Plate Detection*

Pada tahap awal *plate detection* yaitu mengumpulkan data citra gambar plat nomor polisi kendaraan yang diambil dari lalu lintas Kota Medan dan roboflow. Data yang telah dikumpulkan dibagi menjadi data *training*, data *validation*, dan data *testing*.

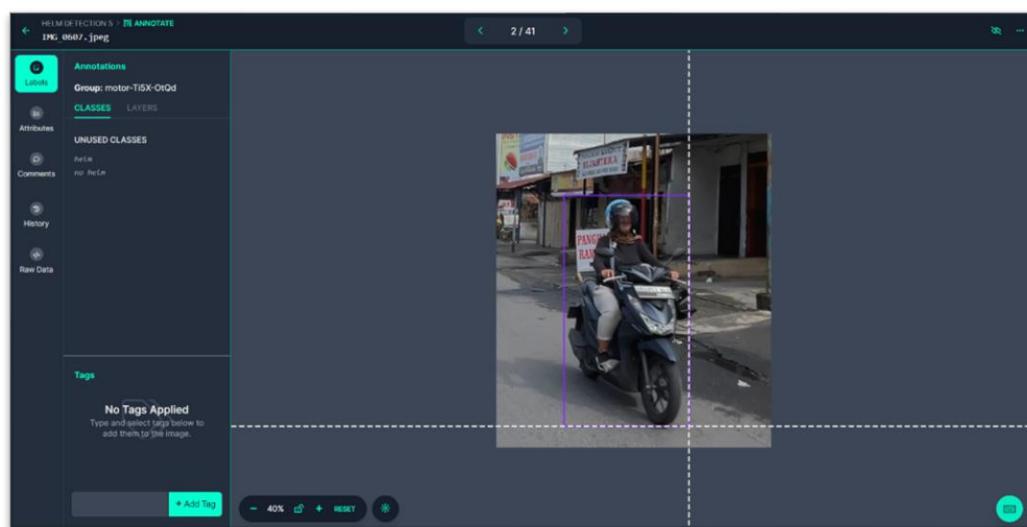
### 3.4. Preprocessing

#### 3.4.1. Motorcyclist Detection

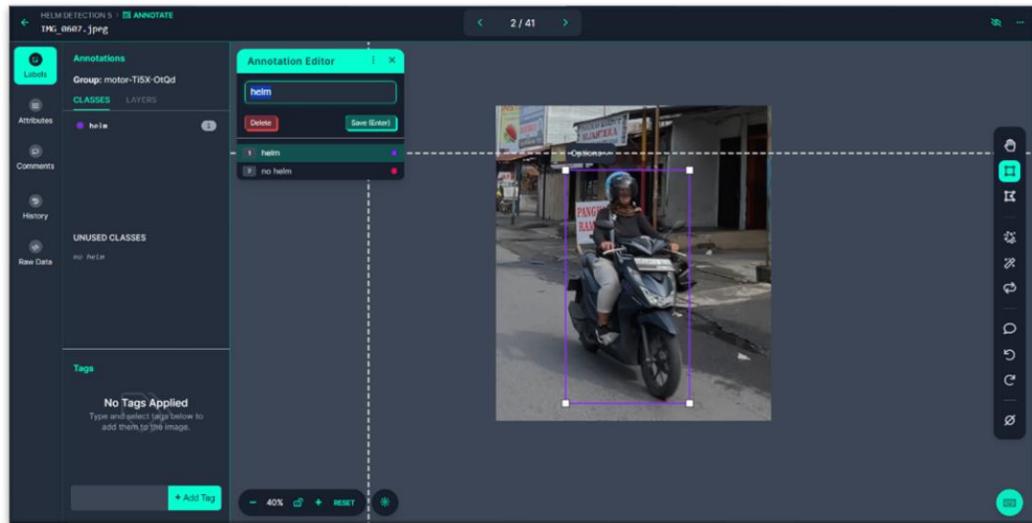
Untuk mencapai hasil yang lebih baik pada tahap berikutnya, data citra akan diproses melalui beberapa proses terlebih dahulu. Beberapa proses *pre-processing* yang akan dilakukan dalam penelitian ini termasuk *image annotation*, *resize*, dan *grayscale*.

##### 3.4.1.1. Image Annotation

*Image annotation* merupakan proses memberikan tanda berupa nama pada data citra yang sudah dikumpulkan untuk selanjutnya digunakan pada pengujian sistem. *Annotation* dilakukan dengan tujuan untuk membedakan jenis atau golongan kelas dari masing-masing citra agar dapat dipelajari oleh model machine learning. Proses pembuatan *bounding box* pada roboflow dapat dilihat pada Gambar 3.4, penulis membuat *bounding box* di sekitar pengendara sepeda motor dan motornya. Setelah penulis membuat *bounding box* maka akan masuk ke tahap pemilihan *class* agar nantinya model akan mempelajari dan dapat membedakan citra pengendara sepeda motor yang menggunakan helm dan yang tidak menggunakan helm, untuk lebih lanjutnya dapat dilihat pada Gambar 3.5.



Gambar 3. 4 Membuat Bounding Box



**Gambar 3. 5 Memilih Class**

Hasil dari pemberian nama atau *labeling* akan disimpan dalam bentuk file.txt isi dari pelabelan tersebut dapat dilihat pada Gambar 3.6. File.txt tersebut berisikan format anotasi sebagai berikut yaitu:

```
<class_id> <x_center> <y_center> <width> <height>
```

```
1 0.58203125 0.62578125 0.06640625 0.06328125
1 0.4078125 0.775 0.09921875 0.07421875
0 0.4109375 0.4234375 0.0875 0.12109375
```

Ln 3, Col 40 | 125 characters | 100% | Unix (LF) | UTF-8

**Gambar 3. 6 Isi File.txt**

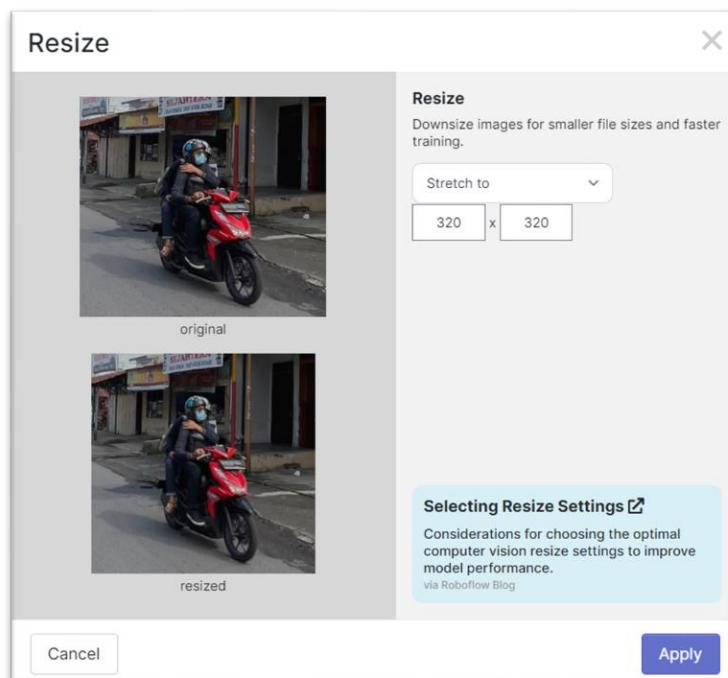
### 3.4.1.2. Resizing

*Resizing* merupakan proses mengubah seluruh ukuran citra yang sudah dikumpulkan menjadi satu ukuran. Tujuan dilakukannya *resizing* yaitu agar model melakukan proses belajar dengan lebih cepat. Pada tahap ini citra di ubah menjadi 320 x 320 pixel. Proses *resizing* dapat dilihat dilihat pada Gambar 3.7.

```

INPUT image l, original_width o, original_height h
DEFINE target_width w, target_height h
SET Scale WHERE
    x_scale = w/o
    y_scale = h/h
SET new_size WHERE
    new_width = o * x_scale
    new_height = h * y_scale
OUTPUT l = new_width x new_height

```



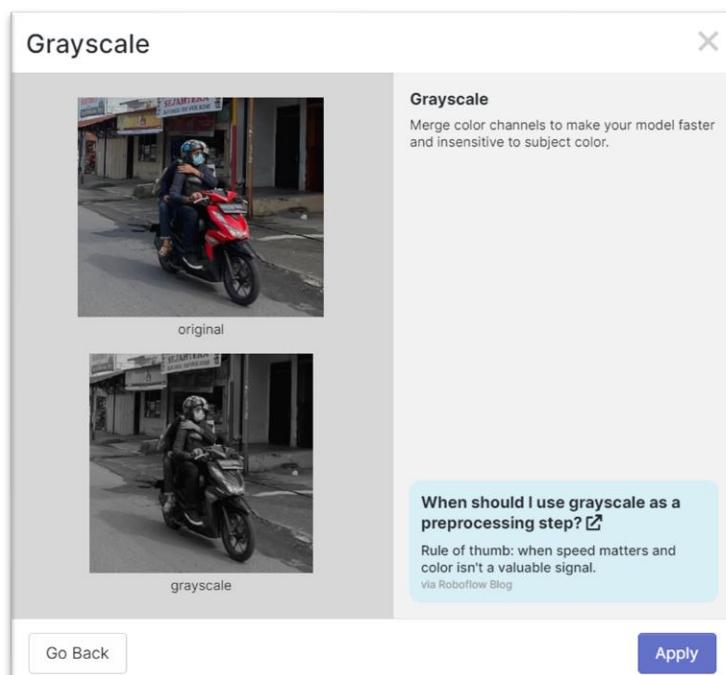
**Gambar 3. 7** Proses *Resizing*

### 3.4.1.3. *Grayscaleing*

*Grayscaleing* merupakan proses merubah citra dari warna lain menjadi nuansa warna monokromatik (hitam dan putih). Tujuan *grayscaleing* yaitu agar model dapat mebedakan warna pada helm yang digunakan oleh pengendara motor dengan mudah. Proses *grayscaleing* dapat dilihat lebih jelas pada Gambar 3.7.

```

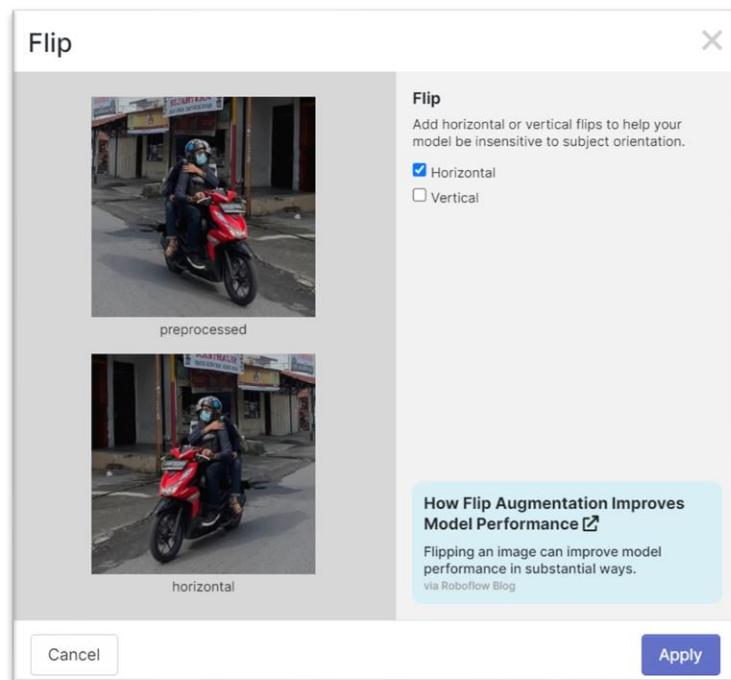
FOR each pixel in Image:
    //Set RGB value of current pixel
    SET Red = Pixel.Red
    SET Green = Pixel.Green
    SET Blue = Pixel.Blue
    //Calculate grayscale value
    SET Gray = (0.299 * Red) + (0.587 * Green) +
                (0.114 * Blue)
    //Set the grayscale value to the new images
    SET Pixel.Red = Gray
    SET Pixel.Green = Gray
    SET Pixel.Blue = Gray
  
```



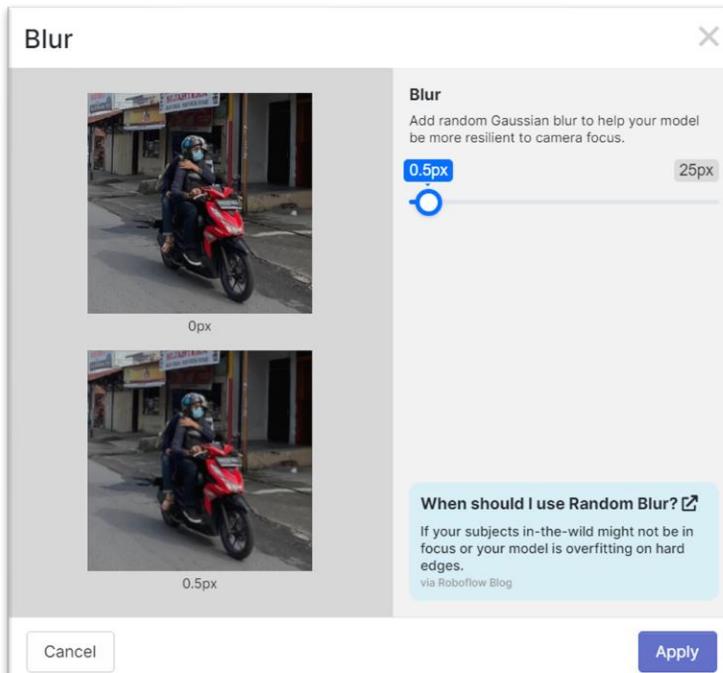
**Gambar 3.8** Proses *Grayscaleing*

#### 3.4.1.4. Augmentation

Pada tahap *augmentation* data yang telah melewati tahap *preprocessing* akan diperbanyak tujuannya agar dataset menjadi lebih besar dan lebih beragam tanpa mengumpulkan data tambahan. Beberapa proses augmentasi yang akan dilakukan pada penelitian ini yaitu flip horizontal dan blur. Dapat dilihat pada Gambar 3.9 terdapat pengendara motor yang sebelumnya berkendara menghadap ke kanan, kemudia setelah melewati proses augmentasi *flip horizontal* pengendara motor tersebut menjadi menghadap ke sebelah kiri. Berikutnya proses *bluring* dapat dilihat pada gambar 3.10 dimana pada bagian atas gambar citra pengendara sepeda motor tidak blur kemudian pada bagian bawah setelah proses augmentasi *bluring* citra pengendara sepeda motor menjadi lebih kabur 0.5px.



Gambar 3.9 Proses *Flip Horizontal*



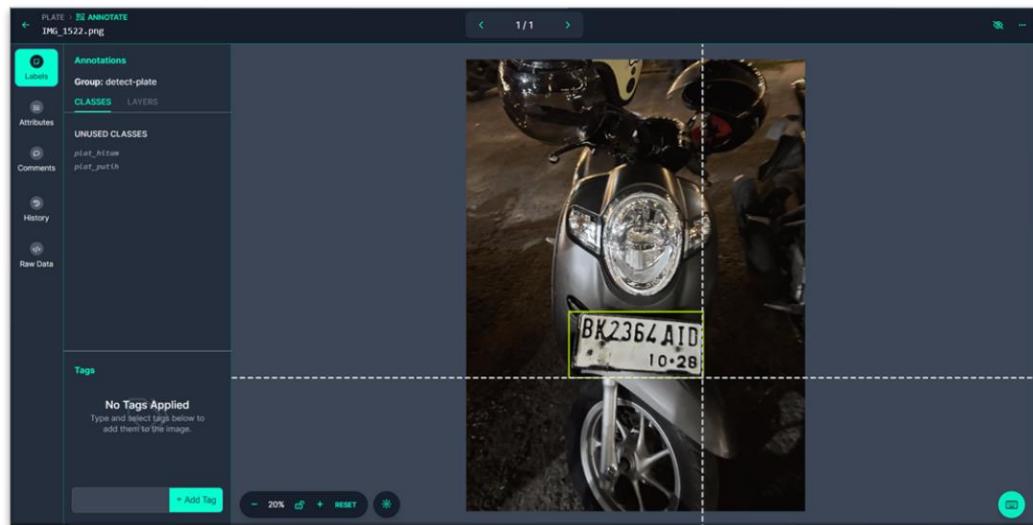
**Gambar 3. 10 Proses *Blur***

#### 3.4.2. *Plate Detection*

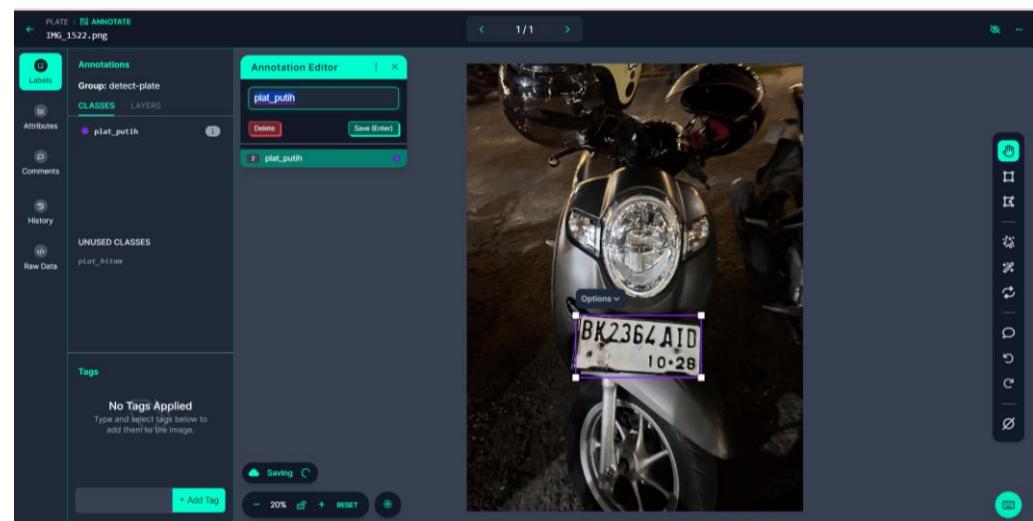
Pada bagian *plate detection*, tahapan pada pre-processing yang akan dilakukan pada penelitian ini, yaitu *image annotation*, *resizing*, *grayscale*, dan *thresholding*.

##### 3.4.2.1. *Image Annotation*

*Image annotation* merupakan proses memberikan tanda berupa nama pada data citra yang sudah dikumpulkan untuk selanjutnya digunakan pada pengujian sistem. *Annotation* dilakukan dengan tujuan untuk membedakan jenis atau golongan kelas dari masing-masing citra agar dapat dipelajari oleh model machine learning. Proses pembuatan *bounding box* pada roboflow dapat dilihat pada Gambar 3.11, penulis membuat *bounding box* di sekitar plat. Setelah penulis membuat *bounding box* maka akan masuk ke tahap pemilihan *class* agar nantinya model akan mempelajari dan dapat membedakan citra pengendara sepeda motor yang menggunakan helm dan yang tidak menggunakan helm, untuk lebih lanjutnya dapat dilihat pada Gambar 3.12.



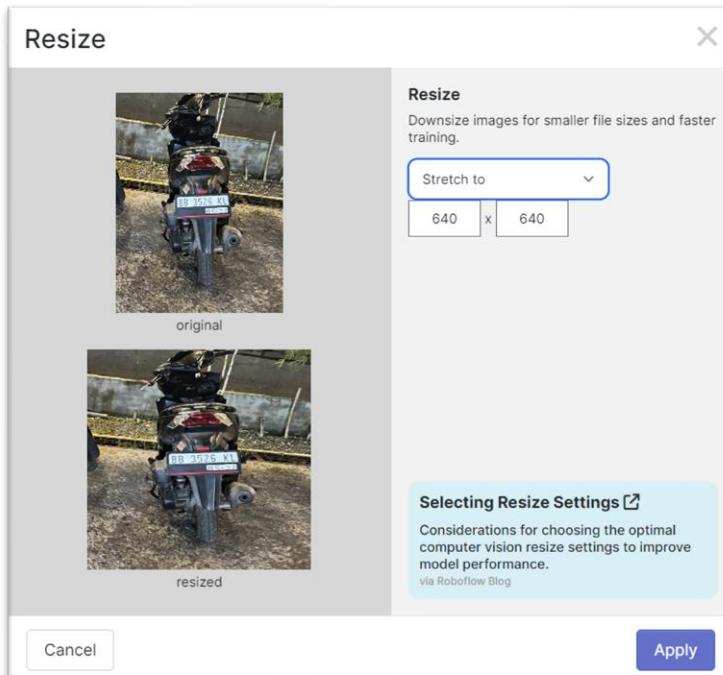
**Gambar 3. 11** Membuat *Bounding Box*



**Gambar 3. 12** Memilih *Class*

#### 3.4.2.2. *Resizing*

*Resizing* merupakan proses mengubah seluruh ukuran citra yang sudah dikumpulkan menjadi satu ukuran. Tujuan dilakukannya *resizing* yaitu agar model melakukan proses belajar dengan lebih cepat. Pada tahap ini citra di ubah menjadi 200 x 100 pixel. Proses resizing dapat dilihat lebih lanjut pada Gambar 3.13.



**Gambar 3. 13 Proses Resizing**

### 3.5. *Detection*

Pada tahap ini, model dilatih untuk mengidentifikasi pengendara sepeda motor yang menggunakan helm dan yang tidak menggunakan helm. Penelitian ini menggunakan algoritma YOLOv8 untuk mendeteksi pengendara sepeda motor dan plat nomor polisi kendaraan sepeda motor.

#### 3.5.1. *Package Installation*

Pada tahap ini, peneliti akan menginstall seluruh *requirement* yang diperlukan untuk pelatihan YOLOv8. Ada dua cara yang dapat digunakan untuk meng-instal YOLOv8 yaitu dengan menggunakan pip sesuai rekomendasi dari pengembang atau secara langsung melalui sumber repositori git ultralytics.

### 3.5.2. Data Configuration

Pada tahap ini, peniliti akan mengimport dataset yang telah melalui tahap *preprocessing* akan diimport menggunakan API Roboflow. Citra untuk training, validasi, dan testing beserta labelnya dan file data.yaml yang merupakan konfigurasi dataset yang digunakan terdapat dalam folder datasets.

```
./helm-detection
└── ./helm-detection/train
    ├── ./helm-detection/train/images
    └── ./helm-detection/train/labels
└── ./helm-detection/valid
    ├── ./helm-detection/valid/images
    └── ./helm-detection/valid/labels
└── ./helm-detection/test
    ├── ./helm-detection/test/images
    └── ./helm-detection/test/labels
└── ./helm-detection/data.yaml
```

Berikut merupakan konfigurasi data pada file data.yaml. Train, val, dan test menunjukkan *path* dari data image, nc menunjukkan jumlah *class*, dan names menunjukkan nama *class* yang digunakan.

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 2
names: ['helm', 'no-helm']
```

### 3.5.3. Train Model

Pada tahap ini citra pengendara motor akan diklasifikasi pada kelasnya menggunakan algoritma You Only Look Once versi 8. YOLOv8 merupakan salah satu algoritma yang mampu mendekripsi objek berdasarkan kemungkinan kelas yang ditetapkan secara *real-time*.

### 3.5.3.1. *Backbone*

Backbone network merupakan tahapan yang mengacu pada feature extraction. Pada tahapan ini backbone network yang digunakan untuk mengekstraksi fitur adalah CSPDarknet53 dimana Backbone terdiri dari dua gabungan yaitu CSP (Cross Stage Partial) dan lapisan dasar convolutional. Berdasarkan perbandingan dengan CSPResNet50 dan EfficientNet-B3, CSPDarknet 53 menghasilkan akurasi yang lebih baik pada pendekalian objek. YOLOv8 menghapus operasi konvolusi setelah pengambilan sampel di struktur PAN, yang mempertahankan performa aslinya sekaligus mencapai model yang ringan. YOLOv8 menggunakan neck yang sama dengan versi sebelumnya yaitu PANet (Path Aggregation Network) yang telah dimodifikasi dengan menambahkannya ke FPN agar mendapatkan akurasi yang lebih baik lagi (Wang, G. et al., 2023).

### 3.5.3.2. *Neck*

Terinspirasi oleh PANet, YOLOv8 dirancang dengan struktur PAN-FPN (Path Aggregation Network with Feature Pyramid Network) di bagian Neck. PANFPN membangun struktur jaringan top-down dan bottom-up guna saling melengkapi informasi posisi dangkal dan informasi semantik yang mendalam melalui perpaduan fitur, sehingga menghasilkan keragaman dan kelengkapan fitur. Pada tahap ini ada penambahan layer dan nantinya akan dibentuk pyramid feature yang berguna untuk menghasilkan prediksi deteksi pada pixel yang kecil sehingga meningkatkan proses segmentasi. Dibandingkan dengan struktur Neck pada model YOLOv5 dan YOLOv7,

### 3.5.3.3. *Head*

Pada tahap ini dihasilkan kotak deteksi yang digunakan untuk mendapatkan koordinat atau bagian dari bounding box objek dan menentukan confidence score pada suatu kelas. Pada tahap ini nantinya akan ada tujuh vektor dimana masing-masing vektor mewakili satu koordinat tertentu. Pada tahap ini juga nantinya akan diimplementasikan fungsi IOU (Intersection Over Union) dan non max suppression untuk menghindari hasil kotak yang tumpang tindih.

### 3.6. Learned Model

Learned model merupakan hasil dari proses pelatihan yang telah dilakukan sebelumnya. Model ini akan digunakan untuk mendeteksi penggunaan helm pada citra pengendara motor sehingga dapat data pelanggaran.

### 3.7. Text Extraction

Text extraction merupakan proses ekstrasi karakter menggunakan EasyOCR. EasyOCR dapat mengubah teks cetakan atau tulisan tangan menjadi teks mesin. Area plat nomor pada gambar diambil dan diubah menjadi format yang sesuai dengan input yang diminta oleh EasyOCR. Kemudian, EasyOCR menjalankan gambar yang telah diubah untuk mengenali karakter optik plat nomor.

#### 3.7.1. Pre-processing

Langkah pertama pada EasyOCR, yaitu diperlukannya *preprocessing* pada gambar hasil deteksi plat nomor polisi kendaraan sebelumnya. Tahapan *preprocessing* yang akan dilakukan yaitu *resizing*, *grayscale*, dan *Binarization*.

##### 3.7.1.1. Resizing

*Resizing* merupakan proses mengubah seluruh ukuran citra yang sudah dikumpulkan menjadi satu ukuran. Tujuan dilakukannya *resizing* yaitu agar model melakukan proses belajar dengan lebih cepat. Pada tahap ini citra di ubah menjadi 200 x 100 pixel. Proses *resizing* dapat dilihat lebih lanjut pada Gambar 3.14 yaitu gambar (a) sebelum proses *resize* dan (b) setelah proses *resize*.

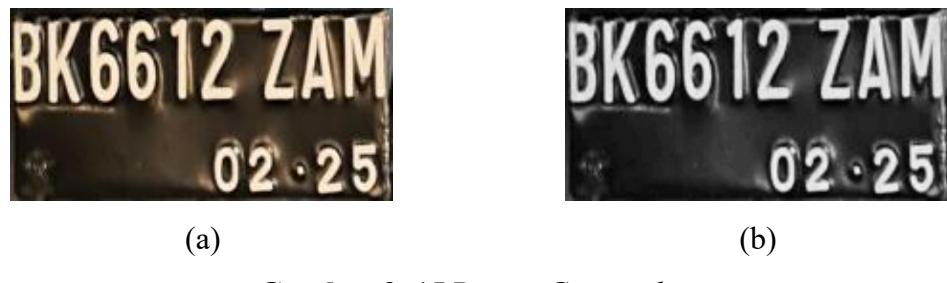


**Gambar 3. 14** Proses *Resizing*

##### 3.7.1.2. Grayscale

*Grayscale* merupakan proses merubah citra dari warna lain menjadi nuansa warna monokromatik (hitam dan putih). Proses *grayscale* dapat dilihat lebih lanjut pada

Gambar 3.15 yaitu gambar (a) sebelum proses *grayscale* dan (b) setelah proses *grayscale*.



**Gambar 3. 15 Proses *Grayscale***

### 3.7.1.3. *Thresholding*

*Thresholding* merupakan proses mengubah citra skala abu-abu menjadi citra biner menggunakan teknik *binarization*, seperti metode Otsu dan *thresholding*. Setelah melakukan beberapa percobaan, nilai ambang batas terbaik yang digunakan adalah 127 yang berarti piksel yang dibawah nilai ambang batas akan diubah menjadi 0 (warna hitam) dan selebihnya akan diubah menjadi 255 (warna putih). Operasi binary thresholding dapat dilihat pada persamaan berikut.

$$dst(x,y) = \begin{cases} maxVal & \text{if } src(x,y) > thresh \\ 0 & \text{otherwise} \end{cases}$$

```

DEFINE threshold_value
// Loop through each pixel in the image
FOR Each Pixel in Image:
    // Compare the pixel value to the threshold
    IF Pixel < threshold_value THEN
        SET Pixel = 0
    ELSE
        SET Pixel = 255
    ENDIF
ENDFOR

```

Proses *thresholding* dapat dilihat lebih lanjut pada Gambar 3.16 yaitu gambar (a) sebelum proses *thresholding* dan (b) setelah proses *thresholding*.



**Gambar 3. 16 Proses Thresholding**

### 3.7.2. *Output*

*Output* dari proses ekstraksi teks plat nomor polisi kendaraan, setelah melakukan *preprocessing* adalah karakter dari plat nomor polisi.

### 3.8. *Deployment*

Pada tahap *deployment* hasil dari proses pelatihan yang berekstensi pt akan dikonversi menjadi file dengan ekstensi tflite dengan TensorFlow Lite. Model yang telah dikonversi kemudian diintegrasikan ke dalam aplikasi *mobile* berbasis Android.

### 3.9. *Output*

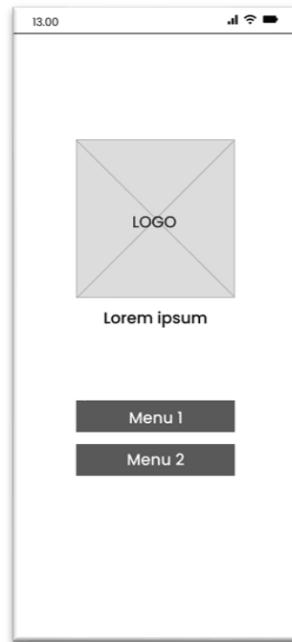
Output merupakan tahap akhir setelah semua proses pada data citra telah dilakukan oleh mesin kemudian memberikan hasil berupa identitas pengendara disertai data pelanggaran penggunaan helm.

### 3.10. *System Interface*

Pada tahap ini, antarmuka pengguna aplikasi dirancang. Tujuan dari tahap ini yaitu untuk memperlihatkan dan menjelaskan ilustrasi dari sistem yang akan dibuat. Halaman yang dirancang yaitu halaman home, halaman deteksi, dan halaman data.

#### 3.10.1. Rancangan Halaman *Home*

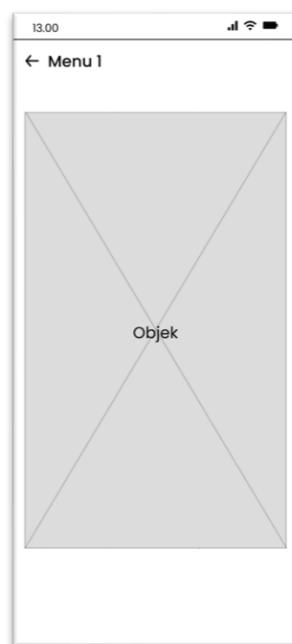
Halaman home merupakan halaman yang akan tampil pada saat aplikasi pertama kali dibuka. Pada halaman home ini akan terdapat logo aplikasi dan dua tombol menu yang akan mengarahkan pengguna ke halamannya masing-masing. Rancangan halaman home dapat dilihat secara jelas pada Gambar 3.17.



**Gambar 3. 17 Rancangan Halaman *Home***

### 3.10.2. Rancangan Halaman *Scan*

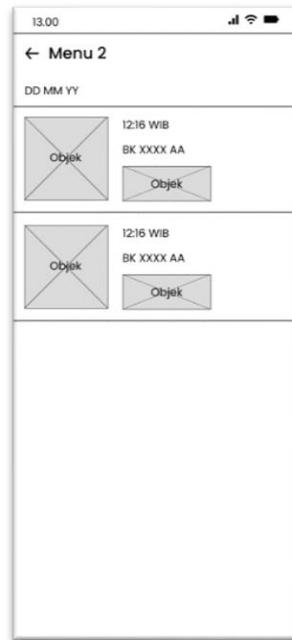
Halaman *scan* merupakan halaman fitur utama yang berfungsi untuk mendeteksi pengendara sepeda motor yang tidak menggunakan helm. Aplikasi akan membuka kamera pada ponsel dan mulai mendeteksi secara real-time. Rancangan halaman home dapat dilihat secara jelas pada Gambar 3.18.



**Gambar 3. 18 Rancangan Halaman *Scan***

### 3.10.3. Rancangan Halaman *Data*

Halaman *data* merupakan halaman yang berisikan data riwayat pengendara yang tidak menggunakan helm. Halaman ini memuat data tanggal, waktu, rekam gambar pengendara yang tinggal menggunakan helm, rekam gambar plat nomor polisi kendaraan dan ekstrasi plat nomor polisi kendaraan. Rancangan halaman home dapat dilihat secara jelas pada Gambar 3.19.



**Gambar 3. 19** Rancangan Halaman Data

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisikan tentang penjelasan proses implementasi dan hasil uji coba algoritma You Only Look Once versi 8 dalam mendeteksi penggunaan helm pada pengendara sepeda motor dan plat nomor polisi kendaraan sepeda motor serta EasyOCR untuk ekstraksi teks pada plat nomor polisi sesuai dengan rancangan sistem pada bab sebelumnya.

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

Pada tahap ini akan dilakukan implementasi algoritma You Only Look Once versi 8 untuk deteksi penggunaan helm pada pengendara sepeda motor dan plat nomor polisi pada kendaraan sepeda motor serta algoritma EasyOCR untuk ekstraksi teks pada plat nomor polisi kendaraan sepeda motor.

##### **4.1.1. Perangkat Keras dan Perangkat Lunak**

Komponen perangkat keras yang digunakan penulis dalam merancang sistem tersebut adalah Laptop Asus Vivobook 14 dengan spesifikasi sebagai berikut:

- 1) Processor: Intel Core i7-1165G7
- 2) RAM: 8 GB
- 3) Storage: 128 MB
- 4) GPU: Nvidia GeForce MX350
- 5) OS: Windows 11 Home SL 64-bit

Komponen perangkat lunak dan *library* yang digunakan untuk membangun sistem yaitu sebagai berikut:

- 1) Google Colab
- 2) Ultralytics 8.0.196

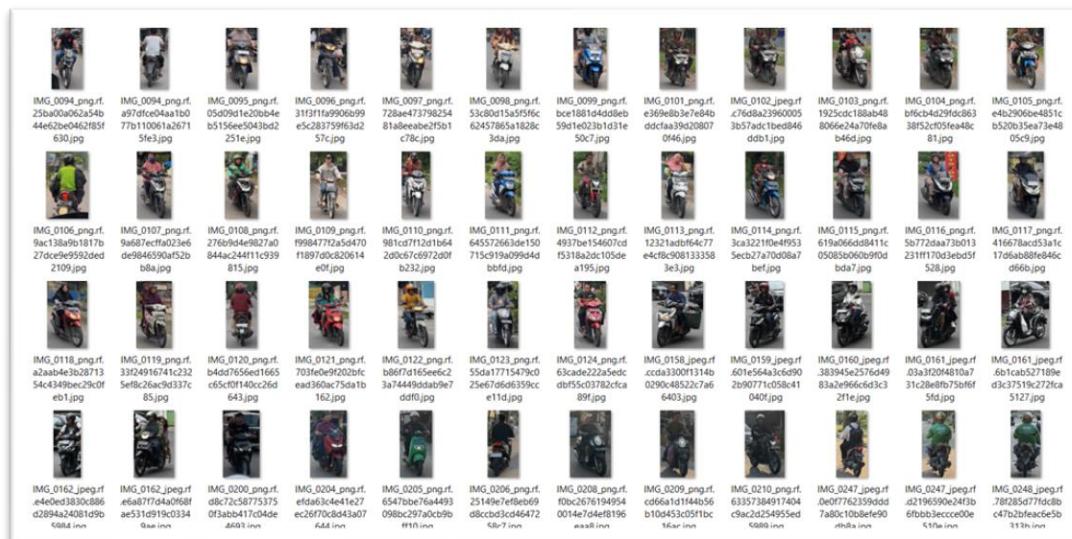
- 3) Bahasa pemrograman Python 3.11.7
- 4) PyTorch 2.3.0
- 5) TensorFlow Lite
- 6) Androis Studio Koala 2024.1.1

Spesifikasi perangkat *mobile* yang digunakan untuk menguji aplikasi secara *real-time* yaitu sebagai berikut:

- 1) OS: Android 13 TKQ1.221013.002
- 2) CPU: Snapdragon 732G
- 3) Storage: 128 GB
- 4) RAM: 6 GB
- 5) Kamera belakang: 108 MP

## 4.2. Implementasi Data

Pada penelitian ini data yang digunakan adalah citra pengendara sepeda motor dan plat nomor polisi kendaraan sepeda motor. Data yang digunakan adalah data yang dikumpulkan secara mandiri menggunakan kamera smartphone dengan ukuran 4K 60FPS. Data diambil disekitaran daerah Kota Medan dilakukan pada siang hari. Beberapa contoh data pengendara sepeda motor yang telah dikumpulkan dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Contoh Data Citra Pengendara Motor

### 4.3. Implementasi Model

Implementasi model menggunakan *software* Visual Studio Code (VS Code). VS Code merupakan *Integrated Development Environment* (IDE) yang mudah digunakan dan banyak digunakan. VS Code sama halnya seperti Google Colab yaitu dapat menjalankan file notebook, namun VS Code menggunakan konfigurasi lokal. Model melakukan pelatihan dengan memanfaatkan *library* dari Ultralytics yang menyediakan YOLOv8.

#### 4.3.1. Sistem Deteksi Pengendara Sepeda Motor

Pelatihan sistem dilakukan dengan menggunakan Google Colab sebagai *tools* dan model *pre-trained* yolov8n.pt, dimana yolov8n merupakan model paling kecil dari algoritma yolov8. Model dilatih sebanyak 150 *epoch* dengan empat jumlah *batch size* yang berbeda yaitu, *batch size* 8, *batch size* 16, *batch size* 32, dan *batch size* 64. *Batch size* menentukan jumlah sampel data yang akan diproses pada satu waktu, makin besar *batch size* maka dapat membuat waktu pelatihan menjadi lebih cepat. Hasil percobaan pelatihan dengan 150 *epoch* dan berbagai nilai *batch size* dapat dilihat pada Tabel 4.1.

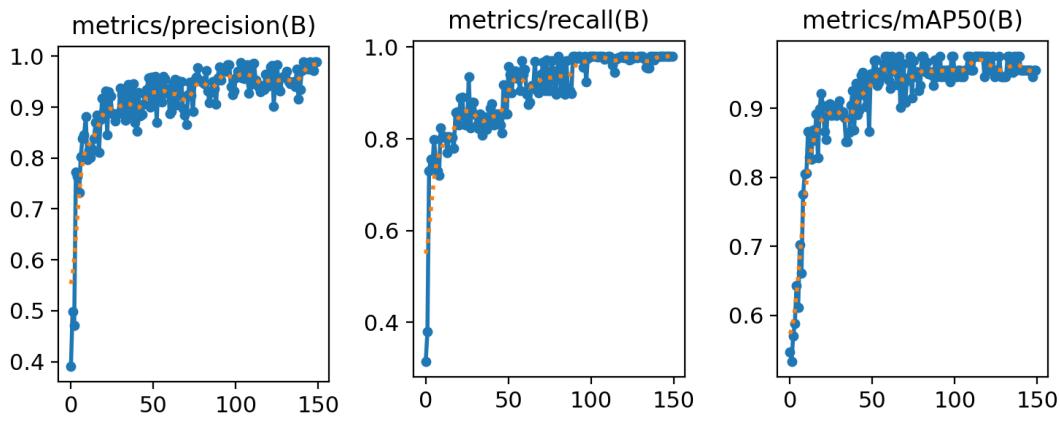
**Tabel 4. 1** Hasil Percobaan 150 *epoch*

Batch size	Precision	Recall	mAP50(B)
8	0.93729	0.99	0.995
16	0.98935	0.98	0.995
32	0.96013	0.99	0.995
64	0.91828	0.75	0.81998

Setelah melewati seluruh proses pelatihan maka hasil terbaik diperoleh pada *batch size* 16 dengan nilai *precision* 0.98935, *recall* 0.98, dan mAP 0,995. Precision berfungsi untuk mengukur proporsi prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model. Pada Gambar 4.2 grafik *precision* meningkat dari sekitar 0,40 ke mendekati 1,00 seiring dengan bertambahnya epoch. Hal ini menunjukkan bahwa model menjadi lebih akurat dalam memprediksi kelas positif seiring dengan pelatihan.

Recall berfungsi untuk mengukur proporsi prediksi positif yang benar dari semua sampel yang benar-benar positif. Grafik *recall* meningkat dari sekitar 0,40 ke

mendekati 1,00 seiring dengan bertambahnya epoch. Ini menunjukkan bahwa model semakin baik dalam menemukan semua sampel positif dalam dataset. mean Average Precision (mAP) pada IoU 0.5 ini merupakan metrik yang sering digunakan dalam tugas deteksi objek, yang mengukur rata-rata precision di berbagai tingkat recall untuk intersection over union (IoU) threshold 0.5. Grafik: mAP50 meningkat dari sekitar 0,70 ke mendekati 0,95 seiring dengan bertambahnya epoch. Ini menunjukkan peningkatan kinerja deteksi objek secara keseluruhan dari model.

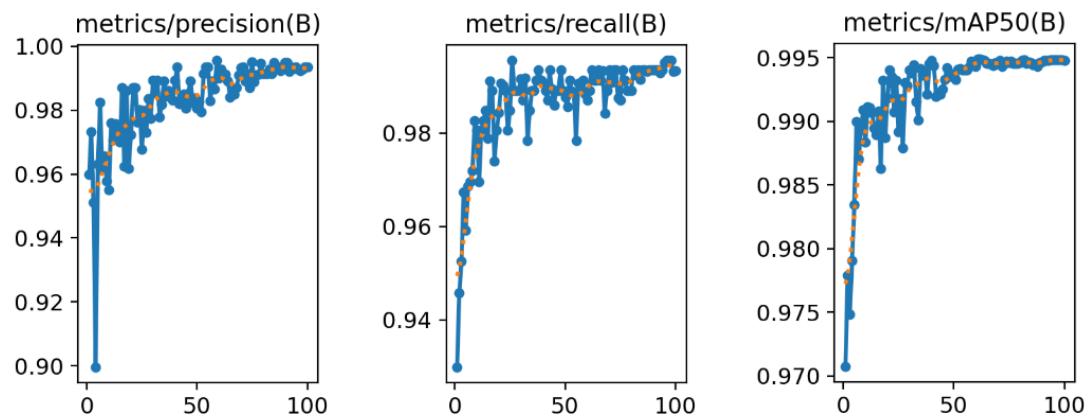


**Gambar 4. 2** Grafik Hasil Percobaan *Batch Size 16*

#### 4.3.2. Sistem Deteksi Plat Nomor Polisi

Pelatihan sistem dilakukan dengan menggunakan Google Colab sebagai tools dan model pre-trained yolov8n. Dapat dilihat pada Gambar 4.3, grafik yang ditampilkan menunjukkan kinerja suatu model pembelajaran mesin dalam tiga metrik yang berbeda: precision, recall, dan mean Average Precision (mAP) pada berbagai epoch, dari 0 hingga 100. Grafik pertama, yang menunjukkan precision, mengindikasikan peningkatan dari sekitar 0,90 ke mendekati 1,00 seiring bertambahnya epoch. Hal ini menunjukkan bahwa model semakin akurat dalam prediksinya.

Grafik kedua, yang menunjukkan recall, juga meningkat dari sekitar 0,94 ke mendekati 0,99, menandakan bahwa model semakin baik dalam menemukan semua sampel positif dalam dataset. Grafik ketiga, yang menunjukkan mAP50, meningkat dari sekitar 0,97 ke mendekati 0,995, mengindikasikan peningkatan kinerja deteksi objek secara keseluruhan. Secara keseluruhan, ketiga grafik ini menunjukkan bahwa model mengalami peningkatan kinerja dalam hal precision, recall, dan mAP50 seiring dengan bertambahnya epoch pelatihan, menandakan bahwa model belajar lebih baik dan menjadi lebih akurat dalam prediksinya seiring dengan waktu pelatihan yang lebih panjang.



**Gambar 4. 3** Grafik Hasil *Batch Size 16*

#### 4.4.Implementasi Antarmuka Aplikasi

##### 4.4.1. Antarmuka Halaman *Home*

Halaman *home* merupakan halaman awal saat aplikasi dibuka. Halaman ini berisikan logo aplikasi dan dua tombol yang akan mengarahkan pengguna ke halaman masing-masing. Tombol *scan* akan mengarahkan pengguna ke halaman deteksi dan tombol *data* akan mengarahkan pengguna ke halaman *data*. Antarmuka halaman *home* dapat dilihat lebih lanjut pada Gambar 4.4.



**Gambar 4. 4** Antarmuka Halaman *Home*

#### 4.4.2. Antarmuka Halaman *Scan*

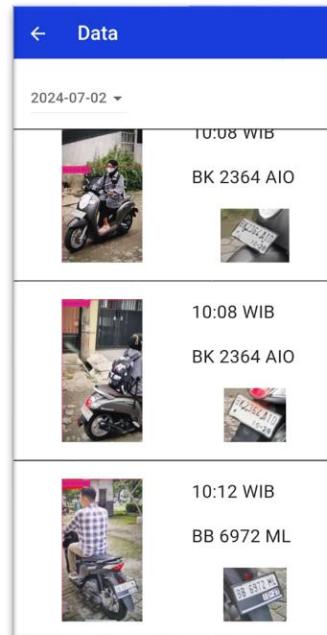
Halaman *scan* akan terbuka jika pengguna menekan tombol *scan* pada halaman *home*. Halaman *scan* merupakan fitur utama pada aplikasi ini. Halaman ini akan mengakses kamera *smartphone* pengguna untuk mendeteksi. Pada halaman ini mempunya tombol deteksi yang fungsinya untuk memunculkan *bounding box* pada kamera. Antarmuka halaman *scan* dapat dilihat lebih lanjut pada Gambar 4.4.



**Gambar 4. 5** Antarmuka Halaman *Scan*

#### 4.4.3. Antarmuka Halaman *Data*

Halaman *data* akan terbuka jika pengguna menekan tombol *data* pada halaman *home*. Halaman ini berikan tanggal, waktu, foto hasil dari deteksi pengendara sepeda motor yang tidak menggunakan helm dan ekstraksi dari plat nomor polisi kendaraan tersebut. Antarmuka halaman *data* dapat dilihat lebih lanjut pada Gambar 4.5



**Gambar 4. 6** Antarmuka Halaman *Data*

#### 4.5.Pengujian Sistem

##### 4.5.1. Sistem Deteksi Pengendara Sepeda Motor

Pada tahap ini, uji coba dilakukan pada model hasil pelatihan dari proses sebelumnya untuk mengetahui kemampuan model dalam mengenali pengendara sepeda motor yang menggunakan helm dan tidak menggunakan helm. Pengujian sistem menggunakan 60 data uji yang dibagi menjadi 30 pada setiap class,. Hasil pengujian sistem dapat dilihat pada Tabel 4.2.

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
1.		No-Helm	No-Helm	Benar

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
2.		No-Helm	No-Helm	Benar
3.		No-Helm	No-Helm	Benar
4.		No-Helm	No-Helm	Benar

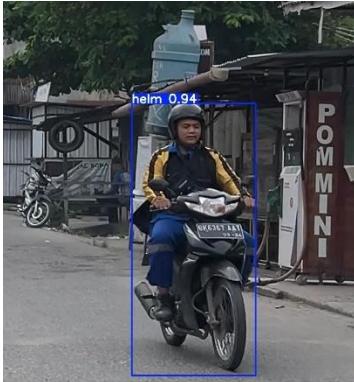
**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
5.		No-Helm	No-Helm	Benar
6.		No-Helm	No-Helm	Benar
7.		No-Helm	No-Helm	Benar

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
8.		Helm	Helm	Benar
9.		Helm	Helm	Benar
10.		No-Helm	Helm	Salah

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
11.		Helm	Helm	Benar
12.		Helm	Helm	Benar
13.		Helm	Helm	Benar

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
14.		Helm	Helm	Benar
15.		Helm dan No-Helm	Helm dan No-Helm	Benar
16.		Helm	Helm	Benar

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
17.		No-Helm	No-Helm	Benar
18.		No-Helm	No-Helm	Benar
19.		Helm	Helm	Benar

**Tabel 4. 2** Hasil Pengujian Sistem *Motorcyclist Detection*

No.	Citra	Aktual	Prediksi	Keterangan
20.		No-Helm	No-Helm	Benar

Berdasarkan hasil pengujian menggunakan 60 data uji, ditemukan beberapa kesalahan pada deteksi. Berikut merupakan hasil perhitungan *confusion matrix* yang dapat dilihat pada Tabel 4.3.

**Tabel 4. 3** *Confusion Matrix Helm Detection*

	TP	FP	FN
Helm	30	0	0
No-Helm	28	2	0
Total	58	2	0

Berikut perhitungan nilai akurasi, *precision*, *recall*, dan *f1-score*:

a. *Precision*

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

$$\begin{aligned}\text{Precision Helm} &= \frac{30}{30 + 0} \\ &= 1\end{aligned}$$

$$\begin{aligned}\text{Precision No - Helm} &= \frac{28}{28 + 2} \\ &= 0.93\end{aligned}$$

b. *Recall*

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

$$\begin{aligned}\text{Recall Helm} &= \frac{30}{30 + 0} \\ &= 1\end{aligned}$$

$$\begin{aligned}\text{Recall No - Helm} &= \frac{28}{28 + 0} \\ &= 1\end{aligned}$$

c. *F1-Score*

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

$$\begin{aligned}F1 - Score Helm &= 2 \times \frac{1 \times 1}{1 + 1} \\ &= 1\end{aligned}$$

$$\begin{aligned}F1 - Score No - Helm &= 2 \times \frac{1 \times 0.93}{1 + 0.93} \\ &= 0.96\end{aligned}$$

d. *Accuracy*

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$\begin{aligned}\text{Accuracy Helm} &= \frac{30 + 0}{30 + 0 + 0 + 0} \times 100\% \\ &= 1\end{aligned}$$

$$\begin{aligned}\text{Accuracy No - Helm} &= \frac{28 + 0}{28 + 0 + 2 + 0} \times 100\% \\ &= 0.93\end{aligned}$$

Setelah melakukan perhitungan terhadap model dengan menggunakan evaluasi matrik, didapatkan precision pada masing-masing *class* adalah 100% dan 93%, *recall* sebesar 100% pada setiap *class* dan *F1-score* sebesar 100% pada *class helm* dan 96% pada *class no-helm*.

Dengan nilai yang dihasilkan dapat disimpulkan model dapat mendekripsi kebenaran objek dengan sangat baik berdasarkan nilai akurasi rata-rata 96%, nilai

*precision* rata-rata 96.5% dan model memiliki sensitivitas yang tinggal dengan memperhatikan nilai *recall* 100%.

#### 4.5.2. Sistem Deteksi Pengendara Sepeda Motor

Pada tahap ini, uji coba dilakukan pada model hasil pelatihan dari proses sebelumnya untuk mengetahui kemampuan model dalam mengenali plat nomor polisi kendaraan sepeda motor. Pengujian sistem menggunakan 60 data uji yang dibagi menjadi 30 pada setiap class, kemudian hasil pengujian sistem dapat dilihat pada Tabel 4.3.

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
1.		plat_putih	Tidak Terdeteksi	Salah
2.		plat_putih	plat_putih	Benar

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
3.		plat_hitam	plat_hitam	Benar
4.		plat_hitam	plat_hitam	Benar
5.		plat_hitam	plat_hitam	Benar

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
6.		plat_hitam	plat_hitam	Salah
7.		plat_hitam	plat_hitam	Benar
8.		plat_putih	plat_putih	Benar

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
9.		plat_hitam	plat_hitam	Benar
10.		plat_putih	plat_putih	Benar
11.		plat_putih	plat_putih	Benar

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
12.		plat_hitam	plat_hitam	Benar
13.		plat_hitam	Tidak Terdeteksi	Salah
14.		plat_hitam	plat_hitam	Benar

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
15.		plat_putih dan plat_putih	plat_hitam dan plat_putih	Salah dan Benar
16.		plat_putih	Tidak Terdeteksi	Salah
17.		plat_hitam	plat_hitam	Benar

**Tabel 4. 4** Hasil Pengujian Sistem *Plate Detection*

No.	Citra	Aktual	Prediksi	Keterangan
18.		plat_putih	plat_putih	Benar
19.		plat_putih	plat_putih	Benar
20.		plat_putih	plat_putih	Benar

Berdasarkan hasil pengujian menggunakan 60 data uji, ditemukan beberapa kesalahan pada deteksi. Berikut merupakan hasil perhitungan *confusion matrix* yang dapat dilihat pada Tabel 4.3.

**Tabel 4. 5 Confusion Matrix Helm Detection**

	TP	FP	FN
Plat Putih	26	1	3
Plat Hitam	28	0	2
Total	54	1	5

Berikut perhitungan nilai akurasi, *precision*, *recall*, dan *f1-score*:

a. *Precision*

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

$$\begin{aligned}\text{Precision Plat Putih} &= \frac{26}{26 + 1} \\ &= 0.96\end{aligned}$$

$$\begin{aligned}\text{Precision Plat Hitam} &= \frac{28}{28 + 0} \\ &= 1\end{aligned}$$

b. *Recall*

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

$$\begin{aligned}\text{Recall Plat Putih} &= \frac{26}{26 + 3} \\ &= 0.89\end{aligned}$$

$$\begin{aligned}\text{Recall Plat Hitam} &= \frac{28}{28 + 2} \\ &= 0.93\end{aligned}$$

c. *F1-Score*

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

$$\begin{aligned}F1 - Score \text{ Plat Putih} &= 2 \times \frac{0.89 \times 0.96}{0.89 + 0.96} \\ &= 0.92\end{aligned}$$

$$\begin{aligned}F1 - Score \text{ Plat Hitam} &= 2 \times \frac{0.93 \times 1}{0.93 + 1} \\ &= 0.96\end{aligned}$$

d. Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$\begin{aligned}\text{Accuracy Plat Putih} &= \frac{26 + 0}{26 + 0 + 1 + 3} \times 100\% \\ &= 0.86\end{aligned}$$

$$\begin{aligned}\text{Accuracy Plat Hitam} &= \frac{28 + 0}{28 + 0 + 0 + 2} \times 100\% \\ &= 0.93\end{aligned}$$

Setelah melakukan perhitungan terhadap model dengan menggunakan evaluasi matrik, didapatkan precision pada masing-masing class adalah 96% dan 100%, recall sebesar 89% dan 93% pada setiap class dan F1-score sebesar 92% pada class plat putih dan 96% pada class plat hitam.

Dengan nilai yang dihasilkan dapat disimpulkan model dapat mendekripsi kebenaran objek dengan sangat baik berdasarkan nilai rata-rata akurasi 89.5%, rata-rata precision 98% dan model memiliki sensitivitas yang tinggal dengan memperhatikan nilai recall rata-rata 91%.

Deteksi plat putih pada kendaraan memiliki tingkat recall yang lebih rendah dibandingkan dengan plat hitam dikarenakan plat putih dengan tulisan hitam memiliki kontras tinggi yang membuatnya mudah dibaca dalam berbagai kondisi pencahayaan. Dalam kondisi cuaca buruk atau pencahayaan rendah, plat putih kurang terlihat hal tersebut dapat mengurangi kesalahan pembacaan oleh model.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Setelah melalui beberapa pengujian deteksi pengendara sepeda motor yang menggunakan helm dan tidak menggunakan helm, juga deteksi plat nomor polisi kendaraan, dapat diambil kesimpulan sebagai berikut:

- 1) Algoritma YOLO yang digunakan sebagai model deteksi pada identifikasi pengguna helm pada pengendara motor dan plat kendaraan berkerja dengan sangat baik.
- 2) Model algoritma You Only Look Once versi 8 memberikan akurasi sebesar 96% dengan nilai precision rata-rata 96.5%, recall 100%, f1-score 98% pada deteksi penggunaan helm pada pengendara.
- 3) Pada deteksi plate kendaraan, model algoritma you only look once versi 8 dapat menghasilkan akurasi sebesar 89.5% dengan nilai rata-rata precision 98%, recall 91%, dan f1-score 94%.
- 4) Kedua model pada system dapat melakukan deteksi secara realtime pada perangkat android.
- 5) Penggunaan sistem kurang optimal pada kondisi kurangnya pencahayaan seperti di sore hingga malam hari, selain itu Jarak pengambilan citra yang terlalu jauh juga dapat mengurangi optimalitas sistem.

#### **5.2. Saran**

Adapun beberapa saran setelah melakukan penelitian yang dapat diberikan pada penelitian selanjutnya yaitu:

- 1) Menggunakan data yang lebih variatif pada aspek tampilan data, jarak serta

sudut pengambilan data serta dengan kuantitas yang lebih banyak agar dapat meningkatkan performa model sehingga hasilnya menjadi lebih akurat.

- 2) Melakukan penelitian dengan metode yang berbeda dan metode baru.
- 3) Mengembangkan sistem dalam platform web yang diintegrasikan dengan kamera cctv lalulintas.

## DAFTAR PUSTAKA

- Aloimonos, Y., & Rosenfeld, A. (1991). Computer Vision. *Science*, 253(5025), 1249–1254. <https://doi.org/10.1126/science.1891713>
- Anderson, M. (2021, June 30). *OCR algorithms: a complete guide*. Itransition. <https://www.itransition.com/blog/ocr-algorithm>
- Aprilino, A., & Amin, I. (2022). Implementasi Algoritma YOLO dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis. *Jurnal TEKNOINFO*, 16(1), 54–59.
- Chellsya, A., Aulia, S., & Hadiyoso, S. (2023). Implementasi Computer Vision Dalam Mendeteksi Pelanggaran Tidak Menggunakan Helm Pada Pengendara Motor. *E-Proceeding of Applied Science*, 9(1), 55–71.
- Cosido, O., Iglesias, A., Galvez, A., Catuogno, R., Campi, M., Teran, L., & Sainz, E. (2014). Hybridization of Convergent Photogrammetry, Computer Vision, and Artificial Intelligence for Digital Documentation of Cultural Heritage - A Case Study: The Magdalena Palace. *2014 International Conference on Cyberworlds*, 369–376. <https://doi.org/10.1109/CW.2014.58>
- Gnanaprakash, V., Kanthimathi, N., & Saranya, N. (2021). Automatic Number Plate Recognition Using Deep Learning. *IOP Conference Series: Materials Science and Engineering*, 1084(1), 012027. <https://doi.org/10.1088/1757-899X/1084/1/012027>
- Illmawati, R., & Hustinawati. (2022). Tampilan YOLOv5 untuk Deteksi Plat Kendaraan di DKI Jakarta. *Jurnal Ilmu Komputer & Agri-Informatika*, 10(1), 32–43.
- Karandish, F. (2022). *The Comprehensive Guide to Optical Character Recognition (OCR)*. Moov Ai. <https://moov.ai/en/blog/optical-character-recognition-ocr>
- Khoiriyah, K., Achmad, F., & Armawan, A. (2023). Deteksi Pengendara Motor Tanpa Menggunakan Helm Dengan Algoritma Deep Learning Yolo. *Jurnal Elektro & Informatika Swadharma (JEIS)*, 03(02), 72–82.
- Miranda, N. D., Novamizanti, L., & Rizal, S. (2020). Convolutional Neural Network pada Klasifikasi Sidik Jari Menggunakan ResNet-50. *Jurnal Teknik Informatika (Jutif)*, 1(2), 61–68. <https://doi.org/10.20884/1.jutif.2020.1.2.18>

- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015a). *You Only Look Once: Unified, Real-Time Object Detection*. <https://arxiv.org/pdf/1506.02640.pdf>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015b). *You Only Look Once: Unified, Real-Time Object Detection*.
- Sankaraiah, Y. R., Harika, K., Manju, C., Anila, K., Reddy, K. R., & Bharathi, G. (2023). Artificial Intelligence Approach for Helmet Detection with Number Plate Extraction Using YOLO. *Journal of Critical Reviews*, 10(03), 1–11.
- Terven, J., & Cordova-Esparza, D. (2024). *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. <https://doi.org/10.3390/make5040083>
- Tun, W. N., Jeon, Y., Kim, S., Nwe Tun, W., Kim, J.-H., & Lee, J.-W. (2020). Safety Helmet and Vest Wearing Detection Approach by Integrating YOLO and SVM for UAV. *KSAS 2020 Spring Conference*. <https://www.researchgate.net/publication/341849696>
- Wiley, V., & Lucas, T. (2018). Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research*, 2(1), 22. <https://doi.org/10.29099/ijair.v2i1.42>
- Zhang, Q., & Zhu, S.-C. (2018). *Visual Interpretability for Deep Learning: a Survey*.