

## Sistem Deteksi Penggunaan Helm Pada Pengendara Sepeda Motor di Indonesia Menggunakan Perbandingan Model YOLOv8 dan RT-DETR

Samuel Orief Rosario<sup>1\*</sup>, Agustinus Aditya Bintara<sup>2</sup>, Muhammad Rifki Zhaki<sup>3</sup>, Rachmat Adi Purnama<sup>4</sup>, Rame Santoso<sup>5</sup>, Veti Apriana<sup>6</sup>

<sup>1-4</sup>Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika, Indonesia

<sup>5</sup>Program Studi Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika, Indonesia

<sup>6</sup>Program Studi Teknik Industri, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika, Indonesia

Email: [samuelorief66@gmail.com](mailto:samuelorief66@gmail.com)<sup>1\*</sup>, [adityabintara04@gmail.com](mailto:adityabintara04@gmail.com)<sup>2</sup>, [muhmmadrijkizhaki@gmail.com](mailto:muhmmadrijkizhaki@gmail.com)<sup>3</sup>,

[rachmat.rap@bsi.ac.id](mailto:rachmat.rap@bsi.ac.id)<sup>4</sup>, [rame.rms@bsi.ac.id](mailto:rame.rms@bsi.ac.id)<sup>5</sup>, [veti.vta@bsi.ac.id](mailto:veti.vta@bsi.ac.id)<sup>6</sup>

\*Penulis korespondensi: [samuelorief66@gmail.com](mailto:samuelorief66@gmail.com)<sup>1</sup>

**Abstract.** *Road safety is an important aspect in reducing accident risks, especially for motorcycle riders. To improve compliance with helmet use, this study compares the performance of two deep learning-based object detection models, namely YOLOv8 and RT-DETR, using a Roboflow dataset consisting of 3,735 images with two classes: with helmet and without helmet. The research process includes data acquisition, preprocessing (512×512 pixels), model training conducted in Visual Studio Code using an Nvidia GTX 1070 Ti GPU with the Ultralytics framework (100 epochs, AdamW optimizer, 0.0005 learning rate, 25 patience), testing on images, videos, and real-time inputs using last.pt, as well as evaluation through precision, recall, mAP, and confusion matrix, followed by implementation of the best algorithm in a local Streamlit web application. The results show that RT-DETR achieved slightly better training performance in terms of mAP50–95, while YOLOv8 performed better during real-world testing with more stable accuracy, particularly for the with helmet class. YOLOv8 reached up to 100% accuracy in video and real-time testing, whereas RT-DETR performed better in the without helmet class, achieving 95% accuracy on image data and up to 100% in video testing. Overall, YOLOv8 was selected as the best model for implementation in the Streamlit-based helmet detection application because it is faster, more stable, and more accurate. This system has the potential to support intelligent ETLE enforcement to enhance traffic safety in Indonesia.*

**Keywords:** Deep Learning; Helmet Detection; RT-DETR; Traffic Safety; YOLOv8

**Abstrak.** Keselamatan berkendara merupakan aspek penting dalam menekan risiko kecelakaan, terutama bagi pengendara sepeda motor. Untuk meningkatkan kepatuhan penggunaan helm, penelitian ini membandingkan performa dua model deteksi objek berbasis *deep learning*, yaitu YOLOv8 dan RT-DETR, menggunakan dataset *Roboflow* berjumlah 3.735 citra dengan dua kelas: *with helmet* dan *without helmet*. Proses penelitian meliputi akuisisi data, *preprocessing* (512×512 piksel), pelatihan model dilakukan di aplikasi *Visual Studio Code* menggunakan GPU *Nvidia GTX 1070 Ti* dengan *Framework Ultralytics* (100 epoch, *AdamW*, learning rate 0.0005, dan 25 patience), pengujian pada *image*, *video*, dan *real-time* menggunakan *last.pt*, serta evaluasi melalui *precision*, *recall*, mAP, serta *confusion matrix* dan terakhir di implementasi di web lokal *streamlit* algoritma yang terbaik. Hasil menunjukkan bahwa RT-DETR memiliki performa pelatihan sedikit lebih baik pada mAP50–95, namun YOLOv8 unggul pada pengujian nyata dengan akurasi lebih stabil, khususnya pada kelas *with helmet*. YOLOv8 mencapai hingga 100% akurasi pada pengujian *video* dan *real-time*, sementara RT-DETR unggul pada kelas *without helmet* untuk data *image* sebesar 95% dan *video* hingga 100%. Secara keseluruhan, YOLOv8 dipilih sebagai model terbaik untuk implementasi pada aplikasi deteksi helm berbasis *Streamlit* karena lebih cepat, stabil, dan akurat. Sistem ini berpotensi mendukung penerapan ETLE cerdas guna meningkatkan keselamatan lalu lintas di Indonesia.

**Kata kunci:** Deteksi Helm; Keselamatan Lalu Lintas; Pembelajaran Mendalam; RT-DETR; YOLOv8

### 1. LATAR BELAKANG

Keselamatan berkendara merupakan tanggung jawab bersama yang harus dijunjung tinggi oleh setiap pengguna jalan. Setiap orang yang berada di jalan raya, baik sebagai pengendara, penumpang, maupun pejalan kaki, memiliki peran dalam menciptakan lingkungan

berlalu lintas yang aman dan teratur. (Purnama et al., 2025) Dominasi sepeda motor sebagai moda transportasi utama di Indonesia memberikan kemudahan dan fleksibilitas tinggi bagi masyarakat dalam beraktivitas sehari-hari. Namun, meningkatnya jumlah pengguna sepeda motor juga berbanding lurus dengan tingginya potensi terjadinya kecelakaan lalu lintas di jalan raya.

Polisi sering mengabaikan pengendara yang melanggar dengan tidak menggunakan helm saat berkendara, yang merupakan salah satu penyebab tingginya angka kecelakaan sepeda motor. (Ilham & Utaminingrum, 2021) Oleh karena itu, deteksi penggunaan helm pada pengendara sepeda motor menjadi sangat penting sebagai upaya meningkatkan keselamatan berkendara. Penerapan teknologi dalam sistem ini berperan untuk mengidentifikasi dan memperoleh informasi secara otomatis mengenai pengendara yang melanggar peraturan, khususnya mereka yang tidak menggunakan helm saat berkendara.

Salah satu langkah yang telah diterapkan oleh pemerintah adalah pemasangan sistem *Electronic Traffic Law Enforcement* (ETLE) atau tilang elektronik di sejumlah lokasi strategis. Sebagai contoh, sistem ETLE di wilayah hukum Polda Metro Jaya telah diperluas secara signifikan di wilayah DKI Jakarta sepanjang tahun 2025, jumlah kamera yang terpasang lebih dari 100 titik. (Daihatsu. A, 2025) Namun demikian, kemampuan kamera tersebut masih terbatas karena belum dapat mendekripsi secara otomatis pengendara sepeda motor yang tidak menggunakan helm.

Untuk mengatasi keterbatasan tersebut, peneliti mengusulkan pemanfaatan teknologi kecerdasan buatan (*Artificial Intelligence/AI*) melalui penerapan model deteksi objek (*object detection*) yang dapat diintegrasikan dengan sistem kamera ETLE. Dengan pendekatan ini, kamera yang terpasang di jalan dapat secara otomatis mengidentifikasi pengendara sepeda motor yang tidak menggunakan helm. Sistem ini memungkinkan penegakan hukum dilakukan dengan lebih efektif, karena proses pendekripsi pelanggaran dapat berlangsung secara *real-time* tanpa harus bergantung sepenuhnya pada pengawasan manual oleh petugas.

Namun, salah satu tantangan utama dalam penerapan sistem pendekripsi helm pada pengendara sepeda motor adalah ukuran kepala pengendara yang relatif kecil dalam citra kamera pengawas, sehingga sering kali sulit terdeteksi dengan akurasi tinggi. Untuk itu, penelitian ini mengusulkan pengembangan sistem atau aplikasi cerdas yang mampu melakukan pendekripsi helm melalui input berupa foto, video, maupun video *real-time* (*Webcam/CCTV*). Sistem ini akan membandingkan kinerja dua model *deep learning*, yaitu YOLOv8 dan RT-DETR, guna menentukan model dengan tingkat akurasi terbaik dalam

mendeteksi penggunaan helm yang mana setelah dibandingkan salah satunya akan diimplementasikan melalui web *local* dengan *streamlit*.

Kedua model ini dikenal unggul dalam mengenali objek berukuran kecil dan kompleks pada citra, dengan YOLOv8 yang berbasis *Convolutional Neural Network* (CNN) berfokus pada kecepatan deteksi *real-time*, sementara *Deformable* RT-DETR yang berbasis *Transformer Architecture* memiliki kemampuan tinggi dalam memahami konteks spasial dan bentuk objek yang bervariasi. Dengan mengintegrasikan hasil penelitian ini ke dalam sistem ETLE, diharapkan kamera pengawas dapat mendeteksi pelanggaran penggunaan helm dengan akurasi yang lebih tinggi dan efisiensi yang lebih baik, sehingga mendukung peningkatan keselamatan lalu lintas di Indonesia.

## 2. KAJIAN TEORITIS

### Studi Terkait

Pada Tabel 1. menunjukkan beberapa penelitian sebelumnya tentang deteksi penggunaan helm pengendara sepeda motor, serta ringkasan dari penelitian tersebut.

**Tabel 1.** Penelitian Terkait.

Referensi	Ringkasan
(Tjitrahardja et al., 2024)	Tahun 2024
	Judul Deteksi Penggunaan Helm Pada Pengendara Motor di Indonesia Menggunakan <i>Deformable</i> DETR
	Metodologi Penelitian ini menggunakan model <i>Deformable</i> DETR ( <i>Deformable Detection Transformer</i> ) untuk mendeteksi penggunaan helm pada pengendara sepeda motor. Dataset diperoleh dari rekaman CCTV publik ( <i>Bali Tower</i> ) yang kemudian dianotasi secara manual menggunakan <i>Roboflow</i> . Eksperimen dilakukan pada beberapa varian model, yaitu DETR, <i>Deformable</i> DETR, <i>Deformable</i> DETR + <i>Iterative Bounding Box Refinement</i> , dan <i>Two-stage Deformable</i> DETR, dengan 50 <i>epoch</i> pelatihan menggunakan GPU <i>Nvidia Tesla P100</i> di <i>Kaggle</i> .
	Kesimpulan Model <i>Two-stage Deformable</i> DETR menghasilkan performa terbaik dengan mAP sebesar 0.611, meningkat 2,1% dibandingkan <i>Deformable</i> DETR biasa (0.590). Meskipun FPS lebih lambat dari DETR konvensional, model ini lebih akurat dalam mendeteksi objek kecil seperti helm. Penelitian ini diharapkan mendukung pengembangan sistem kamera ETLE cerdas untuk penegakan hukum lalu lintas di Indonesia.
	Tahun 2024

(Hidayat & Whardana, 2024)	Judul	Deteksi Pelanggaran Sepeda Motor Menggunakan Algoritma YOLO dan <i>Mean Average Precision</i>
	Metodologi	Penelitian ini menggunakan algoritma YOLO versi 5s dan 8s untuk mendeteksi pelanggaran pengendara sepeda motor, terutama yang tidak menggunakan helm. <i>Dataset</i> diperoleh dari dua sumber, yaitu dataset internet (294 gambar) dan dataset realtime (1.274 gambar). Tiga kelas objek yang digunakan adalah <i>Helmet</i> , <i>NoHelmet</i> , dan <i>Motorcycle</i> . Proses pelatihan dilakukan di <i>Google Colaboratory</i> dengan bantuan GPU dan diimplementasikan menggunakan <i>framework Streamlit (web-based)</i> serta Deepsort untuk menghitung pelanggaran. Evaluasi dilakukan menggunakan <i>Precision</i> , <i>Recall</i> , dan mAP.
	Kesimpulan	Hasil menunjukkan bahwa YOLOv5s dengan <i>dataset</i> internet mencapai akurasi tertinggi sebesar 94%, sedangkan YOLOv5s dan YOLOv8s pada dataset <i>real-time</i> mencapai 93%. Model YOLOv5s dinilai paling optimal untuk implementasi sistem berbasis <i>web</i> dan deteksi pelanggaran helm secara otomatis di lapangan.
(Anggana et al., 2025)	Tahun	2025
	Judul	Deteksi Penggunaan Helm pada Pengendara Sepeda Motor Menggunakan Model YOLOv8 dan <i>Streamlit</i>
	Metodologi	Penelitian ini menggunakan algoritma YOLOv8 untuk deteksi otomatis penggunaan helm pada pengendara motor. <i>Dataset</i> diambil dari <i>Roboflow</i> , dibagi menjadi tiga <i>subset</i> ( <i>train</i> , <i>validation</i> , <i>test</i> ) dengan rasio 70:20:10 dan dua kelas utama: <i>pakai helm</i> dan <i>tanpa helm</i> . Model dilatih di <i>Google Colab</i> (GPU T4) selama 100 <i>epoch</i> dengan resolusi $640 \times 640$ piksel. Sistem kemudian diintegrasikan ke aplikasi <i>web</i> berbasis <i>Streamlit</i> , yang mampu melakukan deteksi pada gambar dan video secara <i>real-time</i> . Evaluasi dilakukan menggunakan metrik <i>mAP@0.5</i> , <i>precision</i> , dan <i>recall</i> .
	Kesimpulan	Model YOLOv8 mencapai <i>mAP@0.5</i> sebesar 88.8%, menunjukkan performa tinggi untuk deteksi helm dalam berbagai kondisi. Sistem berbasis <i>web</i> ini responsif dan akurat, serta berpotensi diterapkan pada pengawasan keselamatan di jalan dan lingkungan kerja industri.

## YOLOv8 dan RT-DETR

*You Only Look Once* (YOLO) adalah algoritma yang digunakan untuk mendeteksi objek secara *real-time*. YOLO menggunakan model unified, di mana *satu convolutional network* memprediksi beberapa kotak pembatas (*bounding box*) dan kemungkinan kelas dalam kotak tersebut secara bersamaan. (Fatkhin & Fadjeri, 2024) YOLOv8 menghadirkan berbagai

peningkatan dan modul baru yang dirancang untuk meningkatkan akurasi serta efisiensi dalam deteksi objek. Arsitektur YOLOv8 merupakan hasil pengembangan dari YOLOv7 dengan sejumlah perbaikan yang signifikan. Salah satu keunggulan utama YOLOv8 adalah kinerjanya yang lebih optimal serta akurasi yang lebih tinggi dalam mendeksi objek. (Iman et al., 2025)

Algoritma deteksi objek *real-time* (DETR) menggunakan arsitektur *transformer* dan mekanisme perhatian untuk mengatasi tantangan deteksi objek. (Khoiriyah, 2023)

### ***Machine Learning dan Deep Learning***

Bidang studi yang dikenal sebagai *machine learning* (ML) berfokus pada desain dan analisis algoritma yang memungkinkan komputer untuk belajar. *Machine learning* juga dapat diprogram, artinya program tersebut dapat menggunakan data untuk membangun model dan membuat keputusan berdasarkan model yang telah dibangun. (Daqiqil. I, 2021)

Bagian pembelajaran mesin yang dikenal sebagai pembelajaran mendalam sangat mirip dengan cara otak manusia menggunakan neuron. Metode ini berkonsentrasi pada pembuatan jaringan saraf palsu dengan beberapa lapisan tersembunyi. Jaringan pembelajaran mendalam yang berbeda termasuk *Multilayer Perceptron* (MLP), *Autoencoder* (AE), *Convolutional Neural Networks* (CNN), dan *Recurrent Neural Networks* (RNN). (Arya et al., 2021)

### ***Convolutional Neural Network (CNN) dan Transformer-based***

Untuk pengolahan gambar dan video, *convolutional neural network* (CNN) terdiri dari lapisan yang disusun dengan susunan *neuron* 3D, yaitu lebar, tinggi, dan kedalaman, dengan lebar dan tinggi merupakan susunan ukuran lapisan, dan kedalaman merupakan jumlah lapisan. (Ariya. C & Lina, 2023)

*Transformer* adalah arsitektur yang didesain untuk mengatasi keterbatasan dalam model berbasis rekursi dan konvolusi dengan mengandalkan mekanisme *self-attention*. Pendekatan ini memungkinkan model untuk menangani berbagai modalitas, seperti teks, gambar, video, dan *audio*, dengan cara yang lebih efisien. (Muazam et al., 2024)

### ***Computer Vision dan Artificial Intelligence***

Jenis kecerdasan buatan yang dikenal sebagai visi komputer adalah jenis kecerdasan buatan yang melatih komputer untuk meniru cara orang melihat, memahami apa yang mereka lihat, dan bertindak berdasarkan informasi yang diproses dan dianalisis. Organisasi yang menggunakan visi komputer dapat mencapai berbagai hasil bisnis, termasuk penyederhanaan proses, peningkatan kinerja, perbaikan pengalaman pelanggan, dan peningkatan keunggulan kompetitif di pasar. (Intel, 2025)

Kecerdasan buatan (AI) adalah program komputer yang bertujuan untuk meniru fitur kecerdasan manusia, seperti pengambilan keputusan logis dan fitur lainnya. AI dapat

digunakan sebagai asisten pembelajaran yang memperkaya dan menyesuaikan pengalaman belajar. (Damayanti et al., 2024)

### **Deteksi Objek**

Deteksi objek adalah metode komputer *vision* yang memungkinkan penemuan objek dalam gambar atau video. Algoritma yang digunakan untuk mendeteksi objek menggunakan pembelajaran mesin mendalam untuk menghasilkan hasil yang baik. (Aningtiyas et al., 2020)

### **Pengolahan Citra Digital**

Pengolahan citra digital, juga dikenal sebagai pengolahan citra digital, adalah bidang ilmu yang mempelajari bagaimana gambar digital dibentuk, diproses, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia. Foto dapat dibagi menjadi dua kategori: analog dan digital berdasarkan bentuk sinyal yang disusunnya. (Ramadhanu & Syahputra, 2022)

### **Helm**

Helm standar merupakan salah satu perlengkapan kendaraan bermotor yang berbentuk topi pelindung kepala bagi pengendara sepeda motor apabila terjadi kecelakaan atau tabrakan sangat dimungkinkan bahwa tubuh pengendara tersebut dapat terpental, dan bisa saja menyebabkan anggota tubuh dan kepala membentur benda keras. Selain sebagai melindungi kepala juga dapat melindungi muka atau wajah para pengendara. (Sarumaha, 2025)

## **3. METODE PENELITIAN**

### **Objek Penelitian**

Objek penelitian ini difokuskan pada deteksi dan analisis penggunaan helm oleh pengendara sepeda motor. Tujuan utama penelitian ini adalah untuk mempelajari pola kepatuhan pengendara terhadap penggunaan helm dengan mempertimbangkan berbagai faktor, seperti lokasi pengambilan data, kondisi lingkungan, serta tingkat kepatuhan terhadap peraturan keselamatan berlalu lintas.



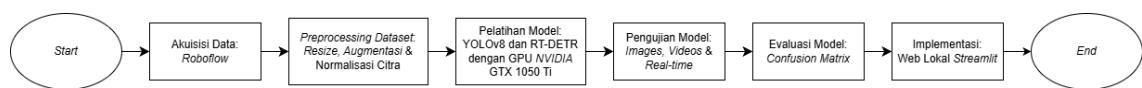
**Gambar 1.** Roboflow.

*Dataset* yang digunakan berasal dari *platform Roboflow* pada Gambar 1, yang menyediakan kumpulan gambar berkualitas tinggi berkaitan dengan penggunaan helm pada pengendara sepeda motor. *Dataset* tersebut mencakup berbagai situasi nyata di jalan raya dan

area publik, dengan variasi kondisi pencahayaan, sudut pandang, serta jenis helm yang digunakan. Data ini kemudian dimanfaatkan untuk melatih dan menguji model deteksi objek, sehingga dapat mengenali dan membedakan antara pengendara yang menggunakan helm dan yang tidak secara akurat.

### Prosedur Penelitian

Prosedur penelitian ini dirancang secara sistematis dan diorganisasikan ke dalam beberapa tahapan, yang divisualisasikan pada Gambar 2. Untuk memastikan keterulangan, validitas, dan transparansi penelitian, setiap tahapan mulai dari akuisisi data, *preprocessing dataset*, pelatihan model, pengujian model, evaluasi model dan implementasi akan dijelaskan secara rinci berikut ini.



**Gambar 2.** Alur Penelitian.

Pada Gambar 2. menunjukkan diagram alur penelitian yang menggambarkan tahapan utama dalam proses pengolahan data deteksi penggunaan helm pada pengendara sepeda motor menggunakan pendekatan YOLOv8 dan RT-DETR. Setiap tahap memiliki peran penting dalam menghasilkan model deteksi yang akurat dan efisien. Adapun penjelasan setiap tahap adalah sebagai berikut:

#### *Akuisi Data*

Untuk mengumpulkan *dataset*, *sample* gambar diambil dari *website Roboflow*. Ini mencakup 3.735 gambar, termasuk validasi 126 gambar, pengujian 63 gambar, dan total 3.546 gambar. *Dataset* ini terbagi dalam dua kelas, kelas 0 *with helmet* dan kelas 1 *Without Helmet*. Baik untuk YOLOv8 maupun RT-DETR, struktur *dataset* terdiri atas dua *folder* utama, yaitu *images* dan *labels*, yang masing-masing menampung *file* gambar serta *file* anotasi dalam format yang sesuai dengan kebutuhan model.

#### *Preprocessing Dataset*

Setelah data terkumpul, dilakukan pra-pemrosesan (*preprocessing*) untuk meningkatkan kualitas dan keragaman *dataset*. Tahapan ini meliputi *resize* (penyesuaian ukuran citra), *augmentasi* (seperti rotasi, flipping, dan perubahan pencahayaan), serta normalisasi citra agar data siap digunakan dalam proses pelatihan model.

#### *Pelatihan Model*

Selanjutnya, proses pelatihan model dilakukan di *Aplikasi Visual Studio Code* dengan GPU *NVIDIA GeForce GTX 1050 Ti* dari PC peneliti untuk mempercepat komputasi. Dua model yang dilatih menggunakan *dataset* yang sama agar hasilnya dapat dibandingkan secara

objektif, sama-sama menggunakan *framework Ultralytics* keduanya dilatih dengan metode *train* bawaan dari *Ultralytics*, dan parameter *train* yang digunakan juga sama. Kelas model yang digunakan berbeda YOLOv8 memakai *yolov8l.pt* sedangkan RT-DETR memakai *rtdetr-l.pt*.

### ***Pengujian Model***

Pada tahap ini, model dijalankan untuk menilai perbandingan akurasi kedua model dalam mendeteksi objek pada citra uji. Hasil prediksi dibandingkan dengan label asli menggunakan *metrik precision, recall, dan mean Average Precision (mAP)*. *Precision* menilai ketepatan deteksi, *recall* mengukur kelengkapan deteksi, sedangkan *mAP* menggambarkan keseimbangan keduanya. Nilai evaluasi ini dihitung otomatis setelah inferensi untuk membandingkan performa YOLOv8 dan RT-DETR yang menggunakan rumus (Baihaqi et al., 2021)

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Seluruh Data}} \times 100\% \quad (1)$$

### ***Evaluasi Model***

*Confusion matrix* adalah alat penting untuk evaluasi model untuk membandingkan kinerja YOLOv8 dan RT-DETR dalam mendeteksi penggunaan helm. Ini memberikan gambaran jelas mengenai jumlah prediksi benar dan salah pada setiap kelas, yang memungkinkan analisis terhadap akurasi, tingkat kesalahan klasifikasi, serta kecenderungan model untuk menghasilkan *false positive* atau *false negative*. Dengan demikian, *confusion matrix* menjadi alat evaluasi penting untuk memastikan bahwa model menggunakan helm dengan benar atau tidak benar.

### ***Implementasi***

Setelah mendapatkan algoritma mana yang terbaik dalam akurasi dan kecepatannya lalu dilakukan implementasi sistem, dengan mengintegrasikan model deteksi helm YOLOv8 ke dalam aplikasi berbasis web menggunakan *Streamlit*. Sistem ini memungkinkan pengguna mengunggah citra uji, menjalankan proses deteksi secara *image*, *video*, dan *real-time*. Setelah mengunggah citra uji hasilnya bisa di *download* dan akan kesimpan ke folder *downloads* di komputer maupun handphone.

## **4. HASIL DAN PEMBAHASAN**

### **Akuisi Data**

Pada tahap awal penelitian, proses akuisisi citra dilakukan untuk memperoleh data visual berupa gambar yang akan digunakan sebagai input dalam sistem deteksi helm. *Dataset*

diperoleh dari *dataset* publik yang tersedia di *platform Roboflow* berikut linknya, (*url*: <https://universe.roboflow.com/bike-helmet/bike-helmet-detection-2vdjo>). *Dataset* yang berisi 3.742 ribu gambar.



**Gambar 3.** Dataset YOLOv8 dan RT-DERT.

## Preprocessing Dataset

Tahap *preprocessing* dilakukan untuk menyiapkan *dataset* sebelum pelatihan model. Setiap citra disesuaikan ukurannya menjadi 512x512 piksel untuk YOLOv8 dan RT-DETR. Proses augmentasi dan normalisasi sudah dilakukan melalui *Roboflow* yang mencakup rotasi, *flipping*, penyesuaian pencahayaan, serta perubahan skala. Tujuannya adalah meningkatkan variasi data, mengurangi risiko *overfitting*, dan menstabilkan proses pelatihan. Dengan tahapan ini, *dataset* menjadi seragam, berkualitas, dan siap digunakan secara optimal oleh kedua model.

## Pelatihan Model

Proses pelatihan model dimulai dengan pembagian *dataset* yang telah dilakukan melalui proses *augmentasi* menjadi tiga subset: pelatihan, validasi, dan uji. Data yang digunakan terdiri dari 3.735 gambar untuk pelatihan, 126 gambar untuk validasi, dan 63 gambar untuk uji.

```
# JALANKAN TRAINING
=====
results = model.train()
data="E:\KULIAH UBST\WAKUL SEMESTER 5\7 - DUMA"
epochs=100,
imgsz=512,
batch=4,
optimizer="AdamW",
lro=0.0005,
patience=25,
augment=True,
ampamp_mode,
project=project_dir,
name="yolov8l",
device=device,
resume=resume_mode,
plots=True,
save=True,
exist_ok=True,
workers=0

# PROSES TRAINING
=====
results = model.train()
data="E:\KULIAH UBST\WAKUL SEMESTER 5\7 - DUMA"
epochs=100,
imgsz=512,
batch=4,
optimizer="AdamW",
lro=0.0005,
patience=25,
augment=True,
ampamp_mode,
project=project_dir,
name="rtdetr-l",
device=device,
resume=resume_mode,
plots=True,
save=True,
exist_ok=True,
Create Jira Issue
workers=0 # ⚠ Fix Windows multiprocessing issue
```

**Gambar 4.** Parameter Training YOLOv8 (kiri) dan RT-DERT (kanan).

Pada Gambar 4. terlihat bahwa proses pelatihan YOLOv8 dan RT-DETR dilakukan dengan konfigurasi parameter yang serupa, seperti jumlah 100 *epoch*, ukuran citra 512x512,

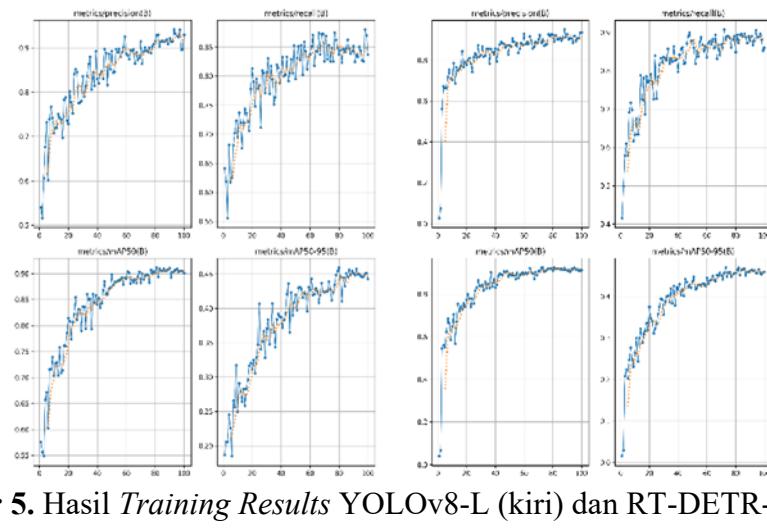
*optimizer AdamW, 0.0005 learning rate, dan 25 patience.* Kedua model juga menggunakan *dataset* yang sama sehingga evaluasi kinerja dapat dilakukan secara objektif. Perbedaan utama terletak pada arsitektur model serta penamaan direktori *output*, di mana YOLOv8 menggunakan nama proyek “yolov8l”, sedangkan RT-DETR menggunakan “rtdetr-l”. Kemudian, setelah di pelatihan hasil dari proses pelatihan kedua model tersebut ditampilkan pada Tabel 2. berikut.

**Tabel 2.** Hasil Perbandingan Pelatihan Model.

Model	Class	Precision	Recall	mAP50	mAP50-95
YOLOv8	<i>All</i>	0.9293	0.8370	0.9006	0.4427
	<i>With Helmet</i>	0.9591	0.8917	0.9320	0.4681
	<i>Without Helmet</i>	0.8993	0.7826	0.8692	0.4189
RT-DETR	<i>All</i>	0.9359	0.8592	0.9123	0.4594
	<i>With Helmet</i>	0.9419	0.8696	0.9187	0.4793
	<i>Without Helmet</i>	0.9397	0.8435	0.9055	0.4430

Pada Tabel 2. menunjukkan hasil evaluasi kinerja model YOLOv8 dan RT-DETR berdasarkan metrik *Precision*, *Recall*, mAP50, dan mAP50-95 pada tiga kategori: *All*, *With Helmet*, dan *Without Helmet*. Secara umum, kedua model menunjukkan performa yang sangat baik, namun RT-DETR sedikit unggul pada sebagian besar metrik. Pada kelas *All*, RT-DETR mencapai *precision* sebesar 0.9359 dan *recall* 0.8592, lebih tinggi dibanding YOLOv8 dengan *precision* 0.9293 dan *recall* 0.8370. Hal serupa terlihat pada nilai mAP50 dan mAP50-95, di mana RT-DETR mencatatkan skor 0.9123 dan 0.4594, sedikit lebih tinggi dibanding YOLOv8 yang memperoleh 0.9006 dan 0.4427.

Pada kelas *With Helmet*, YOLOv8 memiliki *precision* tertinggi sebesar 0.9591, namun RT-DETR menunjukkan mAP50-95 yang lebih baik. Sementara pada kelas *Without Helmet*, RT-DETR kembali menghasilkan nilai *precision*, *recall*, dan mAP yang lebih stabil dibanding YOLOv8. Secara keseluruhan, hasil tersebut menunjukkan bahwa meskipun kedua model memiliki performa yang kompetitif, RT-DETR cenderung memberikan akurasi deteksi yang lebih konsisten, terutama pada situasi variatif dan kelas yang lebih sulit seperti *Without Helmet*.



**Gambar 5.** Hasil *Training Results* YOLOv8-L (kiri) dan RT-DETR-L (kanan).

### Pengujian Model

Sebelum menentukan model terbaik yang akan digunakan pada implementasi melalui web berbasis *Streamlit*, terlebih dahulu dilakukan pengujian menggunakan aplikasi *Visual Studio Code*. Pengujian ini bertujuan untuk membandingkan hasil akurasi antara model YOLOv8 dan RT-DETR pada tiga jenis data yang berbeda, yaitu *image*, *video*, dan *real-time*. Hasil dari pengujian kedua model bisa dilihat pada Gambar 6-8. berikut.



**Gambar 6.** Hasil *Image* (Sumber: Tribun Pontianak).

Pada Gambar 6. terlihat pengujian *image* dengan ukuran *imgsz* 640 pixel dan *conf* 0.20, akurasi yang didapat untuk label *with helmet* YOLOv8 sebesar 0.85 dan RT-DETR sebesar 0.69 dan *without helmet* sebesar 0.89 dan 0.83. Pengujian dilakukan dengan menggunakan 40 citra, citra *class* yang terdeteksi *with helmet* model YOLOv8 sebesar 38 dari 40 dan RT-DETR sebesar 36 dari 40 dengan benar.

$$\text{Akurasi} = \frac{38}{40} \times 100\% = 95\% \quad (1)$$

$$\text{Akurasi} = \frac{36}{40} \times 100\% = 90\% \quad (1)$$

Kemudian, untuk pengujian *class* *without helmet* dilakukan juga dengan menggunakan 40 citra, citra yang terdeteksi model YOLOv8 sebesar 37 dari 40 dan RT-DETR sebesar 39 dari 40 dengan benar.

$$Akurasi = \frac{37}{40} \times 100\% = 92.5\% \quad (1)$$

$$Akurasi = \frac{39}{40} \times 100\% = 97.5\% \quad (1)$$



Gambar 7. Hasil Video (Sumber: ATCS Kota Bandung).

Pada Gambar 7. terlihat pengujian video dengan ukuran yang sama *imgsz* 352 pixel, *tetapi conf berbeda* untuk YOLOv8 menggunakan *conf* 0.60 sedangkan untuk RT-DETR menggunakan *conf* 0.30. Akurasi yang didapatkan untuk label *with helmet* YOLOv8 sebesar 0.89 dan RT-DETR sebesar 0.41 dan *without helmet* sebesar 0.68 dan 0.86. Pengujian *with helmet* dilakukan selama 7 detik dengan 30 fps model YOLOv8 dan RT-DETR sebanyak 210 citra. Citra yang terdeteksi YOLOv8 sebesar 210 dari 210 dan RT-DETR sebesar 210 dari 210 dengan benar.

$$Akurasi = \frac{210}{210} \times 100\% = 100\% \quad (1)$$

$$Akurasi = \frac{210}{210} \times 100\% = 100\% \quad (1)$$

Sedangkan untuk pengujian *without helmet* dilakukan selama 7 detik juga dengan 30 fps model YOLOv8 dan RT-DETR sebanyak 210 citra. Citra yang terdeteksi YOLOv8 20 dari 207 dan RT-DETR 209 dari 210 dengan benar.

$$Akurasi = \frac{207}{210} \times 100\% = 98.57\% \quad (1)$$

$$Akurasi = \frac{209}{210} \times 100\% = 99.52\% \quad (1)$$



Gambar 8. Hasil Real-time.

Pada Gambar 8. terlihat pengujian *real-time* dengan ukuran yang sama *imgsz* 352 *pixel*, tetapi *conf* berbeda untuk YOLOv8 menggunakan *conf* 0.60 sedangkan untuk RT-DETR menggunakan *conf* 0.30. Akurasi yang didapatkan untuk label *with helmet* YOLOv8 sebesar 0.88 dan RT-DETR sebesar 0.75 dan *without helmet* sebesar 0.87 dan 0.84. Pengujian *with helmet* dilakukan selama 7 detik dengan 30 fps model YOLOv8 dan RT-DETR sebanyak 213 citra. Citra yang terdeteksi YOLOv8 sebesar 213 dari 213 dan RT-DETR sebesar 185 dari 213 dengan benar.

$$\text{Akurasi} = \frac{213}{213} \times 100\% = 100\% \quad (1)$$

$$\text{Akurasi} = \frac{185}{213} \times 100\% = 86.90\% \quad (1)$$

Sedangkan untuk pengujian *without helmet* dilakukan selama 7 detik juga dengan 30 fps model YOLOv8 dan RT-DETR sebanyak 213 citra. Citra yang terdeteksi YOLOv8 188 dari 213 dan RT-DETR 181 dari 213 dengan benar.

$$\text{Akurasi} = \frac{188}{213} \times 100\% = 88.26\% \quad (1)$$

$$\text{Akurasi} = \frac{181}{213} \times 100\% = 84.93\% \quad (1)$$

Hasil akurasi dari proses pelatihan kedua model YOLOv8 dan RT-DETR dengan *image*, *video* dan *real-time* tersebut ditampilkan pada Tabel 3. berikut.

**Tabel 3.** Hasil Perbandingan Akurasi.

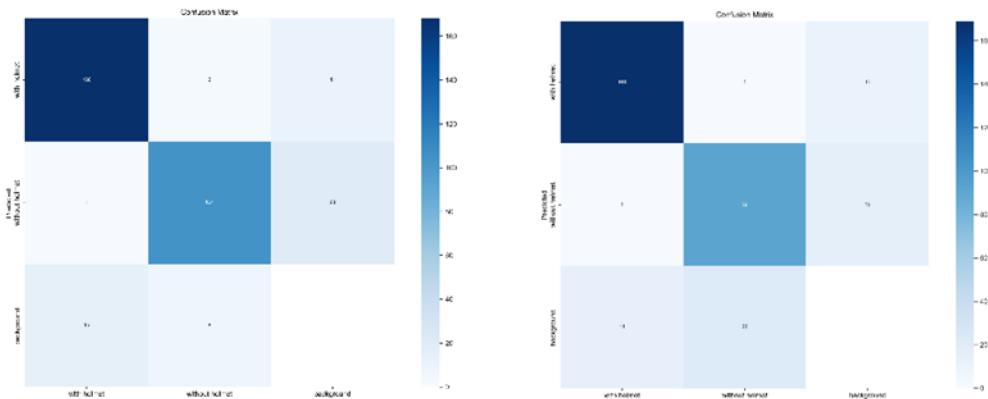
Algoritma	Jenis	Class	Akurasi
YOLOv8	<i>Image</i>	<i>With Helmet</i>	95%
		<i>Without Helmet</i>	92%
	<i>Video</i>	<i>With Helmet</i>	100%
		<i>Without Helmet</i>	98%
	<i>Real-time</i>	<i>With Helmet</i>	100%
		<i>Without Helmet</i>	88%
RT-DETR	<i>Image</i>	<i>With Helmet</i>	90%
		<i>Without Helmet</i>	95%
	<i>Video</i>	<i>With Helmet</i>	100%
		<i>Without Helmet</i>	99%
	<i>Real-time</i>	<i>With Helmet</i>	87%
		<i>Without Helmet</i>	85%

Berdasarkan hasil perhitungan pada Tabel 3, performa kedua model menunjukkan bahwa akurasi deteksi terbaik secara konsisten diperoleh pada pengujian menggunakan *image* dan *video*, di mana YOLOv8 mampu mencapai akurasi hingga 100% pada beberapa kelas, sementara RT-DETR juga menunjukkan hasil tinggi dengan akurasi mencapai 99-100% pada data *video*.

Namun, pada skenario *real-time*, akurasi kedua model cenderung menurun karena adanya dinamika pergerakan objek dan kondisi pencahayaan yang berubah-ubah. YOLOv8 tetap menunjukkan stabilitas lebih baik dengan akurasi 88-100%, sedangkan RT-DETR sedikit lebih rendah dengan kisaran 85-87%. Secara keseluruhan, hasil ini menunjukkan bahwa kedua model bekerja sangat baik pada data statis dan rekaman video, namun membutuhkan penyesuaian lebih lanjut untuk meningkatkan performa pada kondisi *real-time*.

### Evaluasi Model

Selanjutnya, dilakukan evaluasi model YOLOv8 dan RT-DETR yang menggunakan *confusion matrix* yang mencakup metrik *accuracy*, *precision*, dan *recall* untuk menilai ketepatan prediksi dan kemampuan model dalam mendekripsi objek secara benar. *Confusion matrix* juga membantu mengidentifikasi kesalahan klasifikasi, seperti *false positive* dan *false negative*. Selama pelatihan, nilai *loss* dipantau untuk melihat kemampuan model dalam belajar, sementara pengukuran *latency* dan *throughput* digunakan untuk menilai kinerja model pada aplikasi *real-time*. Dengan evaluasi ini, kualitas model dapat dinilai secara menyeluruh dan diketahui apakah model sudah siap diterapkan atau masih perlu peningkatan.



Gambar 9. Hasil *Confusion Matrix* YOLOv8 (kiri) RT-DETR (kanan).

Dapat dilihat pada Gambar 9, kedua *Confusion Matrix* menampilkan performa deteksi helm dari model YOLOv8 dan RT-DETR. Secara umum, kedua model mampu mengklasifikasikan objek *with helmet*, *without helmet*, dan *background* dengan cukup baik. Namun, YOLOv8 menunjukkan jumlah prediksi benar yang lebih tinggi pada kelas *with helmet* dan *without helmet*, ditunjukkan oleh warna biru yang lebih pekat pada diagonal utama. Hal ini menandakan tingkat akurasi yang lebih baik dalam membedakan pengguna helm dan yang tidak memakai helm.

Sementara itu, RT-DETR juga memberikan performa yang stabil, tetapi masih menghasilkan lebih banyak kesalahan pada kelas *background* dan beberapa *misclassification* antar kelas. Dengan demikian, berdasarkan *Confusion Matrix* tersebut, YOLOv8 memiliki

kemampuan deteksi yang lebih konsisten dan presisi dibandingkan RT-DETR, terutama pada skenario yang melibatkan objek serupa namun berbeda kelas. Jadi, benar lebih tepat menggunakan algoritma YOLOv8 untuk mendeteksi objek helm maupun tanpa helm karena lebih akurat, cepat dan efisien.

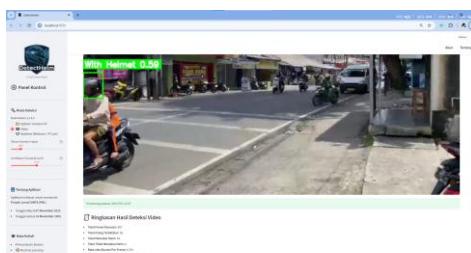
### Implementasi

Setelah hasil evaluasi menunjukkan bahwa YOLOv8 memiliki performa yang lebih unggul dibandingkan RT-DETR, model tersebut kemudian diimplementasikan pada aplikasi web lokal menggunakan *Streamlit* untuk mendukung proses deteksi helm secara *image*, video, dan *real-time* yang mana lokasi penelitiannya dilakukan di depan Kampus UBSI Fatmawati yang berlokasi di Jl. RS. Fatmawati Raya No.24, RT.7/RW.1, Pd. Labu, Kec. Cilandak, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12450. Berikut hasil implementasi Gambar 10-12.



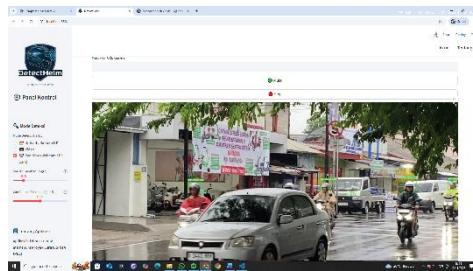
**Gambar 10.** Deteksi *Image*.

Pada Gambar 10. pengujian menggunakan *image*, sistem mampu mengenali objek pengendara dengan sangat baik, termasuk membedakan antara pengendara yang menggunakan helm dan yang tidak menggunakan helm. Deteksi tetap akurat meskipun terdapat lebih dari satu pengendara dalam satu gambar maupun latar belakang yang ramai.



**Gambar 11.** Deteksi *Video*.

Pada Gambar 11. pengujian menggunakan video juga menunjukkan performa yang stabil. Sistem dapat mengikuti pergerakan objek secara berkelanjutan dan memberikan prediksi yang konsisten pada setiap *frame*, sehingga mampu mempertahankan akurasi deteksi dalam situasi dinamis.



**Gambar 12.** Deteksi *Real-time*.

Sementara itu pada Gambar 12, pada pengujian real-time menggunakan webcam, sistem mampu bekerja dengan responsivitas tinggi dan tingkat presisi yang memadai. Deteksi dapat dilakukan secara langsung (*live*) tanpa jeda yang signifikan.

Secara keseluruhan, melalui pengujian pada *image*, *video*, dan *real-time*, sistem terbukti mampu mendeteksi pelanggaran penggunaan helm secara tepat dan andal. Hal ini menunjukkan bahwa sistem layak dijadikan sebagai dasar pengembangan solusi pemantauan otomatis di area dengan lalu lintas padat maupun pada lingkungan industri.

#### 4. KESIMPULAN DAN SARAN

##### Kesimpulan

Berdasarkan hasil penelitian, baik YOLOv8 maupun RT-DETR mampu mendeteksi penggunaan helm pada pengendara motor dengan akurasi rata-rata di atas 90% pada gambar dan video. RT-DETR menunjukkan keunggulan pada metrik pelatihan seperti *precision*, *recall*, dan mAP50-95, terutama pada kelas *without helmet*, berkat arsitektur transformer yang lebih baik dalam memahami konteks spasial. Namun, pada pengujian nyata, YOLOv8 lebih unggul karena lebih stabil, cepat, dan akurat dalam kondisi *real-time* yang dinamis. Hasil *confusion matrix* juga menunjukkan bahwa YOLOv8 memberikan prediksi yang lebih konsisten dengan tingkat kesalahan lebih rendah. Oleh karena itu, YOLOv8 menjadi model paling optimal untuk diimplementasikan pada aplikasi Streamlit, yang terbukti mampu melakukan deteksi pada gambar, video, dan *real-time* secara responsif serta akurat.

##### Saran

Berdasarkan hasil penelitian yang telah dilakukan serta kesimpulan yang diperoleh, terdapat beberapa rekomendasi yang dapat dijadikan acuan untuk penelitian selanjutnya. Penelitian selanjutnya disarankan untuk menambah jumlah *dataset* pelatihan hingga  $>5.000$ - $10.000$  citra agar model memiliki kemampuan generalisasi yang lebih baik. Penggunaan model dengan kapasitas lebih besar seperti *yolov8x.pt* maupun *rtdetr-x.pt* juga direkomendasikan guna meningkatkan akurasi deteksi hingga  $>95\%$ , dengan dukungan perangkat keras

berperforma tinggi seperti GPU RTX >3080/4080 atau layanan *Google Colab Pro*. Selain itu, jumlah *epoch* pelatihan dapat ditingkatkan menjadi 200-500 *epoch* agar model mampu mengenali fitur objek secara lebih optimal. Untuk pengujian *non-realtime*, ukuran *resize* citra 960×960 piksel direkomendasikan atau >512x512, sedangkan untuk *real-time* (CCTV/webcam) disarankan 640×640 piksel agar proses deteksi tetap efisien.

## DAFTAR REFERENSI

- Anggana, M. W., Faizon, A., Munip, A., & Ahyana, A. A. (2025). Deteksi penggunaan helm pada pengendara sepeda motor menggunakan model YOLOv8 dan Streamlit. *Jurnal KTI*, 3(2), 3–6. <https://doi.org/10.26714/jkti.v3i2.18676>
- Aningtiyas, P. R., Sumin, A., & Wirawan, S. (2020). Pembuatan aplikasi deteksi objek menggunakan TensorFlow Object Detection API dengan memanfaatkan SSD MobileNet V2 sebagai model pra-terlatih. *Jurnal Teknologi Informasi*, 19, 421–430.
- Ariya, C., & Lina. (2023). Perancangan deteksi objek pada rak toko menggunakan metode Mask R-CNN. *Jurnal Teknologi*, 8(2), 295–299.
- Arya, K. M., Ajith, K. K., & Engineering, C. (2021). A review on deep learning based helmet detection.
- Astra Daihatsu. (2025). 100 lebih lokasi kamera ETLE di Jakarta: Daftar lengkap dan terbaru tahun 2025. Diakses 9 November 2025, dari <https://www.astra-daihatsu.id/berita-dan-tips/100-lebih-lokasi-kamera-ettle-di-jakarta-daftar-lengkap-dan-terbaru-tahun-2025>
- Baihaqi, K. A., Cahyana, Y., & Buana, U. (2021). Application of convolution neural network algorithm for rice type detection using YOLO v3. *Jurnal Informatika*, 3(2), 272–280.
- Berman, S. (2025). Implementasi pengaturan penggunaan helm bagi pengendara sepeda motor di wilayah Polres Nias Selatan (Studi di Satlantas Polres Nias Selatan). *Jurnal Hukum*, 4(1), 105–118.
- Damayanti, A., Aisyah, A., Riska, Ani, N., & Anna, D. M. (2024). Menilik potensi artificial intelligence bagi masa depan pendidikan global. *Jurnal Pendidikan*, 5(2), 81–91.
- Daqiqil, I. (2021). Machine learning: Teori, studi kasus dan implementasi menggunakan Python. Badan Penerbit Universitas Riau.
- Fatkhin, N., & Fadjeri, A. (2024). Pembelajaran mesin untuk deteksi helm keselamatan menggunakan algoritma YOLOv8. *Jurnal Informatika*, 2, 77–86.
- Hidayat, F. T., & Whardana, A. K. (2024). Deteksi pelanggaran sepeda motor menggunakan algoritma YOLO dan mean average precision. *Jurnal Teknologi*, VIII, 71–79.
- Ilham, I. R., & Utaminingrum, F. (2021). Deteksi helm untuk keamanan pengendara sepeda motor dengan metode CNN (Convolutional Neural Network) menggunakan Raspberry Pi. *Jurnal Teknologi Informasi*, 5(11), 4734–4739.
- Iman, R. M., Rahmadewi, R., & tim lainnya. (2025). Detection of safety helmet usage on workers using You Only Look Once version 8 (YOLOv8). *Jurnal Teknologi*, 27(1), 11–18.
- Intel. (2025). Apa itu computer vision? Diakses 9 November 2025, dari <https://www.intel.co.id/content/www/id/id/learn/what-is-computer-vision.html>

- Khoiriyah, R. (2023). Implementasi algoritma Detection Transformer (DETR) dalam mendekripsi kendaraan di jalan raya. Diakses dari <https://eskripsi.usm.ac.id/files/skripsi/>
- Muazam, S., Kurniawan, Y. I., & Iskandar, D. (2024). Web-based image captioning for images of tourist attractions in Purbalingga using transformer architecture and text-to-speech. *Jurnal Teknologi*, 5(4), 1460–1478.
- Polgan, J. M., Reswara, H. A., Priyatna, B., Hananto, A., & tim lainnya. (2025). Implementasi deteksi objek penggunaan helm dengan metode YOLOv10. *Jurnal Informatika*, 14, 1380–1387.
- Purnama, A., Indra, J., Arum, S., Lestari, P., & Faisal, S. (2025). Deteksi pelanggaran penggunaan helm dengan metode SSD dan arsitektur MobileNetV2. *Jurnal Teknologi*, 7(1).
- Ramadhanu, A., & Syahputra, H. (2022). Pengenalan teknologi pengolahan citra digital (Digital Image Processing) untuk santri di Rahmatan Lil 'Alamin International Islamic Boarding School. *Jurnal Pengabdian*, 3(2), 1239–1244.
- Tjitrahardja, E., Hanif, I. A., Naufal, R. B., & Rahadiani, L. (2024). Deteksi penggunaan helm pada pengendara motor di Indonesia menggunakan Deformable DETR. *Jurnal Informatika*, 2, 41–46.