

# Implementasi Sistem Deteksi Helm Keselamatan Kerja Berbasis YOLOv11 dan *Flask* secara *Real-Time*

Wisnu Wibisono\*, Fersellia

Teknik Informatika, Fakultas Teknik, Universitas Ma'arif Nahdlatul Ulama Kebumen

Email: <sup>1,\*</sup>wiisnuwb@gmail.com, <sup>2</sup>fersellia98@gmail.com

Email Penulis Korespondensi: wiisnuwb@gmail.com\*

Submitted: 18/04/2026; Accepted: 08/06/2026; Published: 30/06/2026

**Abstrak**—Keselamatan dan Kesehatan Kerja (K3) pada sektor konstruksi masih menghadapi permasalahan rendahnya kepatuhan penggunaan alat pelindung diri, khususnya helm keselamatan, yang berkontribusi terhadap meningkatnya risiko kecelakaan kerja. Pengawasan manual dinilai kurang efektif dalam memantau penggunaan helm secara berkelanjutan. Penelitian ini bertujuan untuk mengembangkan sistem deteksi penggunaan helm keselamatan berbasis *computer vision* menggunakan algoritma YOLOv11 yang terintegrasi dengan aplikasi monitoring berbasis web. Sistem dikembangkan menggunakan metode *Rapid Application Development* (RAD). Dataset diperoleh dari *platform* Roboflow sebagai dataset publik yang terdiri dari 19.745 citra dan ditingkatkan menjadi 31.579 citra melalui proses augmentasi, serta tahapan *pre-processing*, anotasi, dan pembagian data. Model dilatih selama 40 *epoch* dengan citra 640×640 piksel dan *batch size* 32, serta dievaluasi menggunakan metrik *precision*, *recall*, dan *mean Average Precision* (mAP). Hasil penelitian menunjukkan bahwa model memperoleh *precision* sebesar 0,881, *recall* sebesar 0,865, dan mAP@0.5 sebesar 0,905, serta mAP@0.5–0.95 sebesar 0,549 yang menunjukkan performa deteksi yang baik. Sistem mampu melakukan deteksi secara *real-time* pada *video streaming* berbasis RTSP serta menampilkan hasil deteksi secara langsung pada setiap *frame*. Pengujian *black-box* menunjukkan seluruh fungsi sistem berjalan dengan baik dan mampu memproses *video streaming* secara kontinu selama monitoring berlangsung. Hasil evaluasi *confusion matrix* menunjukkan model masih berpotensi menghasilkan *false negative* pada kondisi tertentu, khususnya pada objek tanpa helm dengan pencahayaan atau sudut pengambilan gambar yang kurang optimal. Oleh karena itu, pengembangan lebih lanjut diperlukan untuk meningkatkan *robustness* sistem terhadap variasi kondisi lingkungan.

**Kata Kunci:** Computer vision; YOLOv11; deteksi helm; keselamatan kerja; monitoring real-time

**Abstract**—Occupational Safety and Health in the construction sector still faces challenges related to low compliance with the use of personal protective equipment, particularly safety helmets, which contributes to the increasing risk of workplace accidents. Manual supervision is considered less effective in continuously monitoring helmet usage. This study aims to develop a safety helmet detection system based on computer vision using the YOLOv11 algorithm integrated with a web-based monitoring application. The system was developed using the Rapid Application Development (RAD) method. The dataset was obtained from the Roboflow platform as a public dataset consisting of 19,745 images and expanded to 31,579 images through augmentation, as well as preprocessing, annotation, and data splitting stages. The model was trained for 40 epochs using 640×640 pixel images and a batch size of 32, and evaluated using precision, recall, and mean Average Precision (mAP) metrics. The results showed that the model achieved a precision of 0.881, recall of 0.865, mAP@0.5 of 0.905, and mAP@0.5–0.95 of 0.549, indicating good detection performance. The system was able to perform real-time detection on RTSP-based video streaming and display detection results directly on each frame. Black-box testing showed that all system functions operated properly and were capable of processing video streams continuously during monitoring. Confusion matrix evaluation results indicated that the model still had the potential to produce false negatives under certain conditions, particularly for objects without helmets under inadequate lighting or unfavorable camera angles. Therefore, further development is needed to improve the robustness of the system against variations in environmental conditions.

**Keywords:** Computer vision; YOLOv11; helmet detection; workplace safety; real-time monitoring

## 1. PENDAHULUAN

Keselamatan dan Kesehatan Kerja (K3) merupakan aspek penting dalam menjamin perlindungan tenaga kerja, khususnya pada sektor konstruksi yang memiliki tingkat risiko kecelakaan relatif tinggi. Penerapan K3 tidak hanya berperan dalam menekan angka kecelakaan kerja, tetapi juga meningkatkan produktivitas dan efisiensi operasional [1]. K3 bertujuan untuk mencegah dan mengurangi risiko kecelakaan kerja menuju kondisi *zero accident*. Selain itu, K3 dipandang sebagai investasi jangka panjang karena dapat meningkatkan keselamatan serta menurunkan risiko kecelakaan kerja, sekaligus berpengaruh terhadap produktivitas kerja [2]. Secara global, International Labour Organization (ILO) melaporkan lebih dari 250 juta insiden kecelakaan kerja setiap tahun yang dipengaruhi oleh kondisi lingkungan kerja dan perilaku tidak aman [3]. Beberapa penelitian menunjukkan bahwa hingga 80% kecelakaan kerja disebabkan oleh kesalahan manusia, termasuk ketidakpatuhan dalam penggunaan Alat Pelindung Diri (APD) secara benar [4]. Di Indonesia, jumlah kasus kecelakaan kerja pada tahun 2023 mencapai 370.747 kasus dengan peningkatan sebesar 19,7% dibandingkan tahun sebelumnya. Sekitar 32,12% kejadian tersebut disebabkan oleh ketidakpatuhan penggunaan Alat Pelindung Diri (APD) [5]. Hal ini menunjukkan bahwa kepatuhan terhadap prosedur keselamatan masih menjadi faktor utama dalam menekan risiko kecelakaan kerja.

Pada sektor konstruksi, kecelakaan seperti jatuh dari ketinggian, terpeleset, dan tertabrak merupakan insiden yang paling dominan [6]. Karakteristik risiko tersebut menegaskan pentingnya kepatuhan terhadap penggunaan Alat Pelindung Diri (APD) sebagai lapisan perlindungan terakhir dalam sistem pengendalian bahaya. Secara normatif, kewajiban perlindungan tenaga kerja telah ditegaskan dalam Undang-Undang Republik Indonesia

Nomor 1 Tahun 1970 tentang Keselamatan Kerja, yang mewajibkan pemberi kerja menjamin keselamatan setiap pekerja di lingkungan kerja. Namun, tingginya angka kecelakaan menunjukkan adanya kesenjangan antara kerangka regulasi dan implementasi di lapangan. Kondisi ini mengindikasikan bahwa persoalan keselamatan kerja tidak semata-mata terletak pada ketiadaan aturan, melainkan pada efektivitas mekanisme pengawasan dan kepatuhan operasional. Salah satu APD yang paling mendasar adalah helm keselamatan yang berfungsi melindungi pekerja dari risiko cedera kepala. Namun, tingkat kepatuhan penggunaan helm di lapangan masih belum optimal. Pengawasan manual memiliki keterbatasan dalam memantau penggunaan APD secara berkelanjutan, terutama pada proyek dengan jumlah pekerja yang besar. Oleh karena itu, diperlukan sistem pemantauan otomatis yang mampu meningkatkan efektivitas pengawasan secara *real-time* [5].

Perkembangan teknologi *computer vision* memungkinkan penerapan sistem deteksi objek secara otomatis untuk mendukung pengawasan keselamatan kerja. Salah satu algoritma yang banyak digunakan adalah *You Only Look Once* (YOLO) yang dikenal memiliki kemampuan deteksi objek secara *real-time* dengan performa yang baik [7]. YOLO merupakan algoritma deteksi objek berbasis jaringan saraf tiruan yang dirancang untuk melakukan deteksi secara cepat dalam satu tahap pemrosesan. Dalam mekanismenya, citra input dibagi menjadi beberapa area dalam bentuk *grid* berukuran  $S \times S$ , di mana setiap *grid* bertanggung jawab untuk memprediksi keberadaan objek beserta klasifikasinya. Pendekatan ini memungkinkan deteksi objek dilakukan secara efisien dalam satu kali proses. Selain itu, YOLO memiliki kecepatan deteksi yang tinggi, yaitu sekitar 45 hingga 155 *frame* per detik pada pengembangan tertentu, sehingga sesuai untuk sistem monitoring berbasis video yang membutuhkan deteksi cepat dan berkelanjutan [8]. YOLOv11 sebagai versi terbaru dalam keluarga YOLO menghadirkan berbagai penyempurnaan pada arsitektur model, mencakup optimalisasi model, pendekatan *anchor-free detection*, serta penerapan teknik augmentasi modern untuk meningkatkan kemampuan generalisasi dan performa deteksi objek dibandingkan generasi sebelumnya [9].

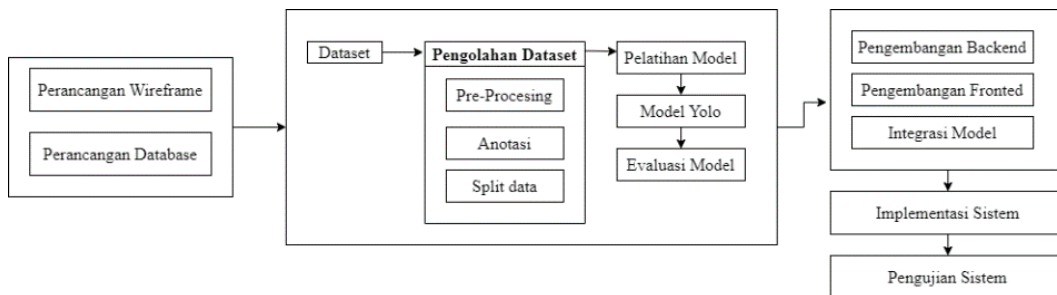
Penelitian terkait mengenai deteksi penggunaan helm keselamatan telah banyak dilakukan dengan berbagai pendekatan berbasis *computer vision*. Penelitian oleh Widodo et al. [10] mengembangkan sistem deteksi helm proyek menggunakan pendekatan *Convolutional Neural Network* dengan algoritma YOLOv2. Sistem yang dibangun mampu melakukan deteksi dalam satu tahap dengan memanfaatkan *anchor box* dan teknik *Non-Max Suppression*, serta menghasilkan nilai *F1-score* sebesar 0,79. Namun, penggunaan YOLOv2 sebagai model generasi awal menunjukkan keterbatasan dalam hal akurasi dan efisiensi dibandingkan dengan arsitektur yang lebih baru. Selanjutnya, penelitian oleh Iman et al. [11] menerapkan algoritma YOLOv8 untuk mendeteksi penggunaan helm secara *real-time* pada pekerja konstruksi. Model yang dikembangkan mampu mencapai tingkat kepercayaan deteksi di atas 90% dan dapat diimplementasikan pada perangkat *edge*. Meskipun demikian, penelitian ini lebih berfokus pada peningkatan performa model dan belum mengkaji integrasi sistem monitoring secara menyeluruh dalam lingkungan operasional. Penelitian lain oleh Reswara et al. [12] menggunakan YOLOv10 untuk mendeteksi penggunaan helm dengan nilai *mAP@0.5* mencapai 99,5%. Hasil tersebut menunjukkan performa model yang sangat tinggi, namun penelitian ini dilakukan pada konteks keselamatan berkendara sehingga kurang merepresentasikan kompleksitas lingkungan konstruksi yang dinamis. Penelitian oleh Pasongko et al. [13] mengembangkan sistem deteksi helm keselamatan menggunakan YOLOv5 dengan dataset sebanyak 694 citra dan menghasilkan nilai *precision* sebesar 0,95 serta *recall* sebesar 0,93. Walaupun memiliki akurasi yang tinggi, penelitian ini masih terbatas pada evaluasi model dan belum mengintegrasikan sistem ke dalam arsitektur monitoring berbasis *real-time*. Penelitian lain menggunakan YOLOv8 dalam pendekatan *machine learning* juga menunjukkan hasil yang cukup baik dengan nilai *precision* 77,7%, *recall* 60,7%, dan *mAP* 68,9% [5].

Meskipun berbagai penelitian telah menunjukkan performa model deteksi yang baik, sebagian besar masih terbatas pada evaluasi berbasis dataset citra statis dan belum merepresentasikan kondisi operasional di lapangan. Pengujian yang dilakukan umumnya hanya berfokus pada akurasi model tanpa mengkaji integrasi sistem secara menyeluruh, seperti proses akuisisi video secara langsung, pemrosesan deteksi secara kontinu pada setiap *frame*, serta penyajian hasil deteksi dalam sistem monitoring. Selain itu, implementasi yang dilaporkan belum mengintegrasikan mekanisme *streaming* berbasis RTSP dan pencatatan pelanggaran secara otomatis ke dalam basis data. Oleh karena itu, terdapat kesenjangan penelitian dalam pengembangan sistem deteksi yang tidak hanya akurat secara model, tetapi juga terintegrasi dalam lingkungan monitoring berbasis web yang berjalan secara kontinu. Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem deteksi penggunaan helm keselamatan berbasis YOLOv11 yang terintegrasi dengan *video streaming* berbasis RTSP serta aplikasi monitoring berbasis web, serta mengevaluasi kinerja sistem secara menyeluruh dalam skenario operasional. Pengembangan sistem dilakukan menggunakan pendekatan *Rapid Application Development* (RAD), yaitu metode pengembangan perangkat lunak yang menekankan proses iteratif dan *prototyping* untuk menghasilkan sistem dalam waktu yang relatif lebih singkat [14]. Kontribusi utama penelitian ini adalah integrasi proses deteksi objek, akuisisi video, dan pencatatan pelanggaran ke dalam satu *platform monitoring* yang berjalan secara kontinu. Berbeda dengan penelitian sebelumnya yang umumnya menggunakan pendekatan *multi-class detection*, penelitian ini memfokuskan deteksi pada dua kondisi utama, yaitu pekerja yang menggunakan helm dan tidak menggunakan helm. Kinerja sistem dievaluasi menggunakan metrik *precision*, *recall*, dan *mean Average Precision* (mAP), serta analisis *confusion matrix*.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini menggunakan model pengembangan *Rapid Application Development* (RAD) sebagai kerangka kerja dalam membangun sistem deteksi helm keselamatan kerja. RAD merupakan model pengembangan perangkat lunak yang menekankan siklus pengembangan singkat dengan pendekatan iteratif [15]. Model ini dipilih karena bersifat iteratif dan mampu mendukung pengembangan sistem secara cepat, sehingga sesuai untuk sistem monitoring berbasis *real-time*. RAD terdiri dari beberapa tahapan utama, yaitu *Requirement Planning*, *User Design*, dan *Construction* [16]. Tahap *Requirement Planning* dilakukan untuk mengidentifikasi kebutuhan sistem. Tahap *User Design* mencakup perancangan arsitektur, antarmuka, dan basis data. Selanjutnya, tahap *Construction* meliputi pengolahan dataset, pelatihan model YOLOv11, serta pengembangan aplikasi berbasis web menggunakan *Flask*.

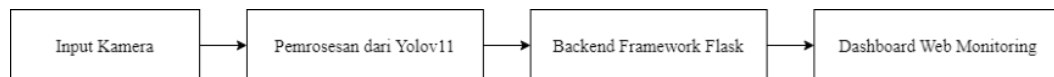


**Gambar 1.** Tahapan Penelitian

Gambar 1 menunjukkan alur penelitian yang dimulai dari tahap perancangan sistem, meliputi perancangan *wireframe* dan basis data. Selanjutnya dilakukan pengumpulan dan pengolahan dataset melalui tahapan *pre-processing*, anotasi, dan pembagian data. Dataset yang telah diproses digunakan untuk pelatihan dan evaluasi model menggunakan metrik *precision*, *recall*, dan *mean Average Precision* (mAP). Model yang memenuhi kriteria kemudian diintegrasikan ke dalam sistem melalui pengembangan *backend* dan *frontend*. Tahap akhir adalah implementasi serta pengujian sistem untuk memastikan fungsionalitas berjalan sesuai kebutuhan.

### 2.2 Arsitektur Sistem

Arsitektur sistem deteksi helm keselamatan kerja pada penelitian ini dirancang sebagai sistem terintegrasi berbasis *computer vision* dan aplikasi web untuk mendukung monitoring secara *real-time*. Sistem menerima input video dari kamera melalui protokol RTSP yang kemudian diproses menjadi *frame* citra. Setiap *frame* dianalisis menggunakan model YOLOv11 untuk mendeteksi objek pekerja beserta status penggunaan helm keselamatan. Hasil deteksi selanjutnya dikelola oleh *backend* berbasis *Flask* dan ditampilkan pada *dashboard* web dalam bentuk video *real-time* yang telah dilengkapi *bounding box* dan informasi status penggunaan helm.

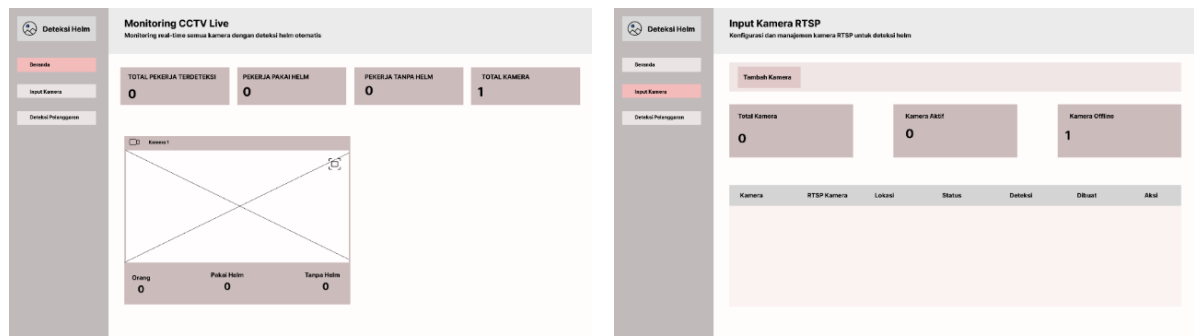


**Gambar 2.** Arsitektur Sistem

### 2.3 Perancangan Sistem

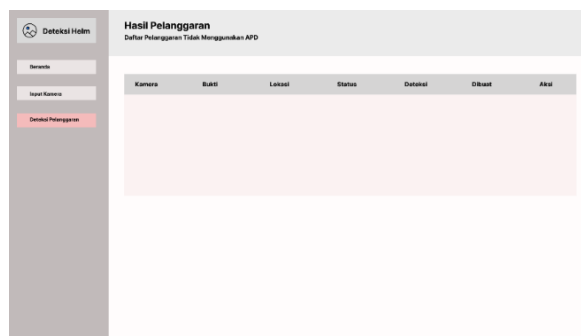
#### 2.3.1 Perancangan Wireframe (Antarmuka)

Perancangan antarmuka dilakukan untuk menggambarkan alur interaksi pengguna dengan sistem deteksi helm secara visual. *Wireframe* sistem terdiri dari tiga halaman utama, yaitu *dashboard monitoring*, halaman input kamera, dan halaman deteksi pelanggaran. *Dashboard* berfungsi sebagai tampilan utama yang menampilkan *video streaming* secara *real-time* beserta informasi jumlah pekerja terdeteksi, pekerja yang menggunakan helm, dan tidak menggunakan helm. Halaman input kamera digunakan untuk mengelola sumber video berbasis RTSP, termasuk penambahan dan pengaturan status kamera. Sementara itu, halaman deteksi pelanggaran menampilkan hasil deteksi pekerja yang tidak menggunakan helm dalam bentuk tabel sebagai dokumentasi kejadian. Perancangan ini bertujuan untuk memastikan sistem mudah digunakan serta mampu menyajikan informasi secara jelas dan terstruktur guna mendukung proses monitoring keselamatan kerja.



(a) Antarmuka Beranda

(b) Antarmuka Input Kamera



(c) Antarmuka Deteksi Pelanggaran

**Gambar 3.** Perancangan *Wireframe*: (a) Beranda, (b) Input Kamera dan (c) Deteksi Pelanggaran

### 2.3.2 Perancangan Database

Perancangan basis data dilakukan menggunakan pendekatan relasional dan diimplementasikan dengan PostgreSQL. PostgreSQL merupakan sistem manajemen basis data relasional berorientasi objek yang dikenal stabil, fleksibel, serta memiliki dukungan fitur yang lengkap [17]. Selain itu, PostgreSQL sebagai basis data *open source* mampu mendukung pengembangan aplikasi berbasis *web* maupun *desktop* [18]. Dalam penelitian ini, struktur basis data disederhanakan menjadi dua tabel utama, yaitu *cameras* dan *violations*. Tabel *cameras* digunakan untuk menyimpan informasi perangkat kamera, seperti nama, lokasi, alamat RTSP, serta status operasional. Sementara itu, tabel *violations* digunakan untuk mencatat data pelanggaran yang terdeteksi, meliputi referensi kamera (*camera\_id*), lokasi file hasil deteksi, dan waktu kejadian. Relasi antar tabel direpresentasikan melalui atribut *camera\_id* pada tabel *violations* yang mengacu pada id pada tabel *cameras*, sehingga memungkinkan satu kamera memiliki banyak data pelanggaran. Struktur ini dirancang untuk menjaga keterkaitan data serta mendukung proses monitoring secara berkelanjutan. Untuk meningkatkan efisiensi akses data, digunakan indeks pada atribut *camera\_id* dan *detected\_at*.

**Tabel 1.** Struktur Tabel *Cameras*

Nama	Tipe Data	Keterangan
id	int	Primary key, identitas unik kamera
name	varchar(255)	Nama kamera
location	varchar(255)	Lokasi kamera
rtsp_url	text	Alamat RTSP sebagai sumber video
description	text	Deskripsi tambahan kamera
status	varchar(20)	Status kamera
last_violation	boolean	Status pelanggaran terakhir
last_online	timestamp	Waktu terakhir kamera aktif
created_at	timestamp	Waktu data dibuat

**Tabel 2.** Struktur Tabel *Violations*

Nama	Tipe Data	Keterangan
id	int	Primary key data pelanggaran
camera_id	int	Relasi ke tabel <i>cameras</i>

image_path	text	Lokasi file gambar hasil deteksi
detected_at	timestamp	Waktu terjadinya pelanggaran

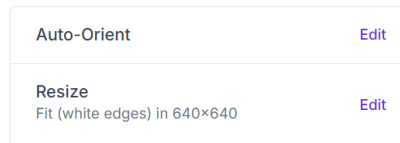
## 2.4 Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini menggunakan data sekunder berupa dataset citra digital. Dataset yang digunakan diperoleh dari *platform Roboflow Universe* melalui proyek *Hard Hats Computer Vision Model* [19], yang menyediakan kumpulan citra terkait deteksi penggunaan alat pelindung diri (APD). Dataset tersebut terdiri atas citra pekerja dengan variasi penggunaan helm keselamatan, perbedaan sudut pengambilan gambar, kondisi pencahayaan, serta latar belakang lingkungan kerja yang beragam. Jumlah dataset awal sebanyak 19.745 citra, kemudian dilakukan proses augmentasi sehingga jumlah data meningkat menjadi 31.579 citra. Data difokuskan pada dua kelas utama, yaitu *Pakai\_Helm* dan *Tidak\_Pakai*. Dataset yang digunakan bersifat publik sehingga memungkinkan penelitian direproduksi pada kondisi yang serupa.

## 2.5 Pengolahan Data

### 2.5.1 Pre-Processing

*Pre-processing* dilakukan untuk menyiapkan data agar sesuai dengan input model sebelum pelatihan. Tahap ini meliputi normalisasi citra dengan ukuran  $640 \times 640$  piksel menggunakan metode *fit (white edges)* serta penerapan *auto-orient* untuk menjaga konsistensi orientasi gambar. Selain itu, nilai piksel disesuaikan mengikuti standar input YOLOv11 untuk meningkatkan stabilitas proses pelatihan.



**Gambar 4.** Normalisasi Data

### 2.5.2 Pelabelan dan Anotasi

Setelah tahap *pre-processing*, dataset citra dilanjutkan ke proses pelabelan dan anotasi objek menggunakan *platform Roboflow*. Proses anotasi dilakukan dengan teknik *bounding box*, di mana setiap objek pekerja dalam citra ditandai sesuai dengan kelasnya. Hasil anotasi disimpan dalam format *.txt* mengikuti standar YOLO. Pelabelan difokuskan pada dua kelas utama, yaitu *Pakai\_Helm* dan *Tidak\_Pakai*.



**Gambar 5.** Proses Anotasi Data

Visualisasi pada Gambar 5 menunjukkan bahwa dalam satu citra dapat terdapat lebih dari satu objek dengan kategori berbeda.

COLOR ?	CLASS NAME	COUNT ↻
<span style="color: green;">●</span>	Pakai_Helm	42,428
<span style="color: magenta;">●</span>	Tidak_Pakai	12,966

**Gambar 6.** Jumlah Anotasi per Kelas

Gambar 6 menunjukkan distribusi jumlah anotasi pada setiap kelas dalam dataset. Dataset awal yang berjumlah 19.745 citra menghasilkan total anotasi sebanyak 42.428 untuk kelas *Pakai\_Helm* dan 12.966 untuk kelas *Tidak\_Pakai*. Jumlah anotasi tersebut merepresentasikan banyaknya objek dalam seluruh citra, sehingga satu citra dapat memuat lebih dari satu objek.

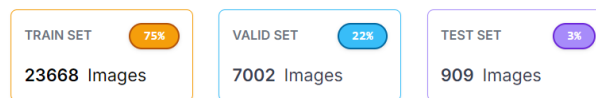
<b>Flip</b> Horizontal, Vertical	<a href="#">Edit</a>
<b>Rotation</b> Between -15° and +15°	<a href="#">Edit</a>
<b>Blur</b> Up to 2.5px	<a href="#">Edit</a>

**Gambar 7.** Augmentasi Dataset

Selanjutnya, dilakukan proses augmentasi untuk meningkatkan variasi data citra. Augmentasi dilakukan menggunakan *platform* Roboflow dengan teknik seperti *horizontal flip*, *vertical flip*, rotasi, dan blur. Setelah proses augmentasi, jumlah dataset meningkat menjadi 31.579 citra.

### 2.5.3 Pembagian Dataset

Pembagian dataset merupakan tahapan penting dalam pengembangan model *machine learning* untuk memastikan model dapat dilatih dan dievaluasi secara objektif menggunakan data yang belum pernah dilihat sebelumnya. Dalam penelitian ini, pembagian dataset dilakukan menggunakan *platform* Roboflow dengan proporsi 75% data pelatihan, 22% data validasi, dan 3% data pengujian. Data pelatihan digunakan untuk melatih model YOLOv11, data validasi digunakan untuk memantau performa model selama proses pelatihan, sedangkan data pengujian digunakan untuk mengevaluasi kinerja model secara objektif pada data yang tidak terlibat dalam proses pelatihan.



**Gambar 8.** Pembagian Dataset

## 2.6 Pelatihan Model

Pelatihan model deteksi helm keselamatan kerja dilakukan menggunakan algoritma YOLOv11 yang dirancang untuk deteksi objek secara *real-time* [20]. Proses pelatihan dilakukan pada Google Colab dengan dukungan GPU untuk mempercepat komputasi. Dataset yang telah melalui tahap *pre-processing*, anotasi, augmentasi, dan pembagian data digunakan dalam format YOLO yang terdiri dari data *training* dan *validation*. Model diinisialisasi menggunakan bobot awal *yolo11n.pt* dan dilatih selama 40 *epoch* dengan ukuran citra 640 × 640 piksel serta *batch size* 32. Pemilihan parameter ini mempertimbangkan keseimbangan antara kecepatan pelatihan dan performa model. Proses optimisasi dilakukan menggunakan algoritma Adam yang mampu menyesuaikan *learning rate* secara adaptif sehingga membantu mempercepat konvergensi model [21], [22]. Dataset dikonfigurasi melalui file *data.yaml* dengan mekanisme *cache disk* untuk meningkatkan efisiensi pembacaan data selama pelatihan.

```

from ultralytics import YOLO

model = YOLO("yolo11n.pt")

model.train(
    data=f"{dataset.location}/data.yaml",
    epochs=40,
    imgsz=640,
    batch=32,
    optimizer="Adam",
    cache="disk"
)

```

**Gambar 9.** Konfigurasi Pelatihan Model

## 2.7 Implementasi Sistem

Tahap implementasi dilakukan dengan mengintegrasikan model deteksi ke dalam sistem monitoring berbasis *web* menggunakan *framework Flask* yang dibangun menggunakan bahasa pemrograman Python. *Flask* adalah *web framework* berbahasa Python yang memberikan kebebasan pada *developer* untuk mengembangkan suatu *website* dan tergolong dalam *micro-framework* karena tidak memerlukan *tools* tambahan untuk menggunakannya [23]. *Flask* dipilih karena merupakan *micro-framework* yang ringan dan fleksibel sehingga memudahkan pengembangan aplikasi web serta integrasi dengan berbagai pustaka Python yang digunakan dalam pengolahan data dan pengembangan model kecerdasan buatan [24]. Sistem menerima input berupa *video streaming* dari kamera melalui protokol RTSP yang kemudian diproses menjadi *frame* citra. Setiap *frame* dianalisis menggunakan model YOLOv11 untuk mendeteksi objek pekerja dan mengklasifikasikan penggunaan helm keselamatan. Hasil deteksi berupa *bounding box* dan label kelas selanjutnya diproses oleh *backend* dan ditampilkan pada *dashboard* web secara *real-time*. Pada sisi *frontend*, sistem menampilkan *video streaming* yang telah dilengkapi dengan hasil deteksi objek sehingga pengguna dapat melakukan monitoring penggunaan helm secara langsung.

## 2.8 Evaluasi Model

Evaluasi performa model dilakukan untuk mengukur kinerja deteksi objek menggunakan *metrik precision, recall, dan mean Average Precision (mAP)*. *Precision* dan *recall* digunakan untuk menilai ketepatan serta kelengkapan deteksi objek, sedangkan *mAP* digunakan sebagai metrik utama karena mampu merepresentasikan akurasi deteksi secara keseluruhan berdasarkan kurva *precision-recall*. Perhitungan *mAP* didasarkan pada nilai *Intersection over Union (IoU)* antara *bounding box* prediksi dan *ground truth*. Pada penelitian ini digunakan ambang *IoU* sebesar 0,5 (*mAP@0.5*) untuk menentukan apakah suatu deteksi dikategorikan sebagai benar atau salah. Selain itu, evaluasi juga didukung menggunakan *confusion matrix* untuk menganalisis performa prediksi label pada objek yang terdeteksi, khususnya dalam membedakan kelas Pakai\_Helm dan Tidak\_Pakai. *Confusion matrix* memetakan hasil prediksi terhadap label sebenarnya sehingga menghasilkan nilai *true positive (TP)*, *false positive (FP)*, *false negative (FN)*, dan *true negative (TN)*. Berdasarkan nilai tersebut, dihitung metrik tambahan berupa *accuracy* dan *F1-score* untuk memberikan gambaran yang lebih komprehensif terhadap performa model dalam mengklasifikasikan objek yang telah terdeteksi.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<p><b>TP</b> (True Positive)</p>	<p><b>FP</b> (False Positive) <small>Type I Error</small></p>
	0 (Negative)	<p><b>FN</b> (False Negative) <small>Type II Error</small></p>	<p><b>TN</b> (True Negative)</p>

Gambar 10. Confusion Matrix

Berdasarkan nilai *true positive (TP)*, *false positive (FP)*, *false negative (FN)*, dan *true negative (TN)*, metrik evaluasi dihitung sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

Untuk *mean Average Precision (mAP)*, digunakan perhitungan rata-rata dari *Average Precision (AP)* pada setiap kelas, yang diperoleh dari luas kurva *precision-recall*:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \tag{3}$$

## 2.9 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *black-box testing* untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan. Pengujian dilakukan pada beberapa skenario, seperti proses penerimaan *video streaming*, deteksi objek secara *real-time*, serta tampilan hasil deteksi pada *dashboard* web. Hasil pengujian

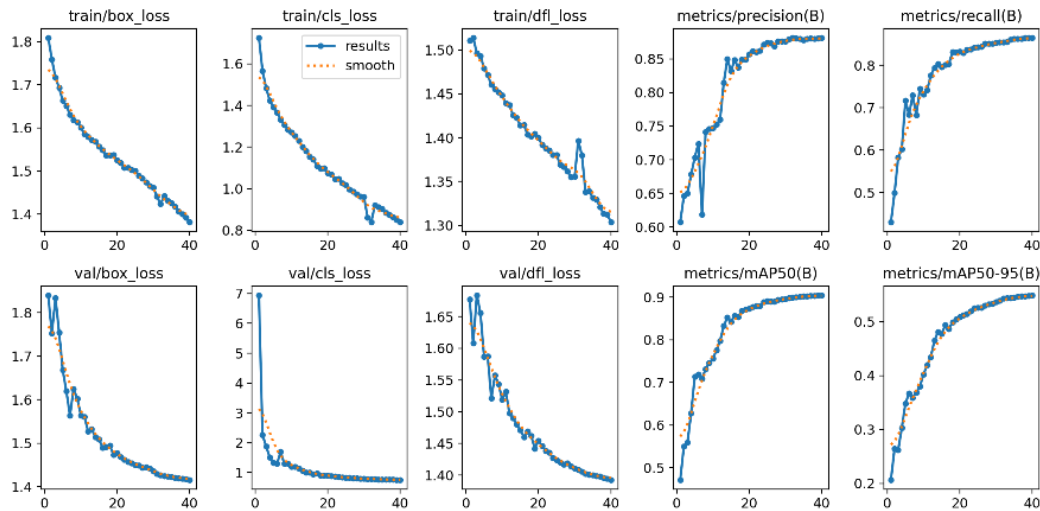
digunakan untuk memastikan bahwa sistem mampu berjalan dengan baik dan menghasilkan output yang sesuai dengan tujuan penelitian.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil Pelatihan dan Evaluasi Model

##### 3.1.1 Hasil Pelatihan Model

Pelatihan model dilakukan menggunakan algoritma YOLOv11 dengan parameter yang telah ditentukan. Proses pelatihan berlangsung selama 40 *epoch* dengan ukuran citra 640×640 piksel dan *batch size* 32.



**Gambar 11.** Hasil Pelatihan Model (*loss*, *precision*, *recall*, dan *mAP*)

Gambar 11 menunjukkan tren penurunan *loss* serta peningkatan metrik evaluasi *precision*, *recall*, dan *mean Average Precision* (mAP). Nilai *loss* yang terdiri dari *box loss*, *classification loss*, dan *distribution focal loss* (DFL) pada data *training* dan *validation* mengalami penurunan yang konsisten, menunjukkan peningkatan kemampuan model dalam melakukan lokalisasi dan klasifikasi objek. Kesesuaian antara *training loss* dan *validation loss* mengindikasikan bahwa proses pelatihan berlangsung stabil tanpa indikasi *overfitting*. Seiring dengan penurunan *loss*, nilai *precision* meningkat hingga sekitar 0,88, yang menunjukkan kemampuan model dalam meminimalkan kesalahan deteksi (*false positive*), sementara *recall* mencapai sekitar 0,86 yang menunjukkan peningkatan kemampuan model dalam mendeteksi objek secara lebih lengkap. Nilai mAP@0.5 mencapai sekitar 0,90, yang menunjukkan performa deteksi yang baik pada ambang *Intersection over Union* (*IoU*) moderat. Namun, nilai mAP@0.5:0.95 yang berada pada kisaran 0,55 menunjukkan bahwa akurasi penempatan kotak pembatas (*bounding box localization*) pada objek secara presisi tinggi masih terbatas dan perlu ditingkatkan. Keterbatasan tersebut umumnya dipengaruhi oleh variasi ukuran objek, sudut pengambilan gambar, serta kondisi objek yang tertutup sebagian (*occlusion*). Secara keseluruhan, model telah menunjukkan performa yang stabil dan kemampuan generalisasi yang baik sehingga layak untuk diimplementasikan pada sistem deteksi berbasis *real-time*. Untuk analisis lebih lanjut, evaluasi performa model per kelas disajikan pada Tabel 3.

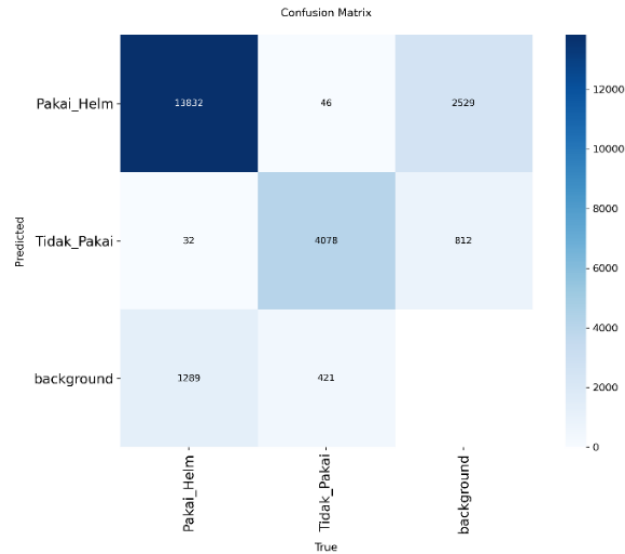
**Tabel 3.** Hasil Pembelajaran Model per Kelas

Kelas	<i>Precision</i>	<i>Recall</i>	mAP@0.5	mAP@0.5:0.95
Pakai_Helm	0.895	0.869	0.922	0.530
Tidak_Pakai	0.867	0.862	0.888	0.567
Semua	0.881	0.865	0.905	0.549

Berdasarkan Tabel 3, performa model pada kedua kelas menunjukkan distribusi yang relatif seimbang. Kelas *Pakai\_Helm* memiliki nilai *precision* dan *recall* sedikit lebih tinggi dibandingkan *Tidak\_Pakai*, yang menunjukkan bahwa model lebih konsisten dalam mengenali objek dengan karakteristik visual yang lebih jelas. Nilai mAP@0.5 yang tinggi pada kedua kelas menunjukkan kemampuan deteksi yang baik pada ambang *IoU* moderat, sementara nilai mAP@0.5:0.95 yang lebih rendah menunjukkan bahwa akurasi lokalisasi objek masih perlu ditingkatkan. Secara keseluruhan, model tidak menunjukkan bias signifikan terhadap salah satu kelas.

### 3.1.2 Evaluasi Model

Evaluasi model dilakukan menggunakan *confusion matrix* untuk menganalisis distribusi hasil prediksi sebagaimana ditunjukkan pada Gambar 14.

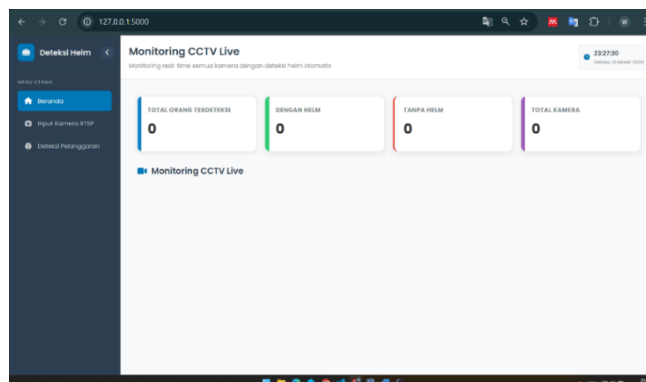


Gambar 12. Confusion Matrix

*Confusion matrix* menunjukkan bahwa sebagian besar prediksi berada pada diagonal utama, yang mengindikasikan bahwa model mampu mengklasifikasikan objek dengan benar. Pada kelas *Pakai\_Helm*, jumlah *true positive* mencapai 13.832, sedangkan pada kelas *Tidak\_Pakai* mencapai 4.078, yang menunjukkan performa deteksi yang baik pada kedua kelas. Namun, masih terdapat kesalahan prediksi berupa *false negative*, di mana objek tidak terdeteksi oleh model dan diklasifikasikan sebagai *background*. Kesalahan ini lebih dominan pada kelas *Tidak\_Pakai*, yang mengindikasikan adanya kesulitan dalam mendeteksi objek pada kondisi tertentu seperti pencahayaan rendah, resolusi objek kecil, atau *occlusion*. Hal ini menunjukkan bahwa model memiliki sensitivitas yang lebih tinggi terhadap objek yang memiliki ciri visual mencolok, seperti penggunaan helm keselamatan, dibandingkan dengan kepala tanpa atribut pendukung. Selain itu, terdapat pula kesalahan klasifikasi antar kelas dalam jumlah terbatas, yang menunjukkan bahwa model masih mengalami ambiguitas pada kondisi visual tertentu. Secara keseluruhan, meskipun masih terdapat keterbatasan pada kondisi tertentu, model telah menunjukkan tingkat keandalan yang cukup untuk diimplementasikan dalam sistem monitoring berbasis *real-time*.

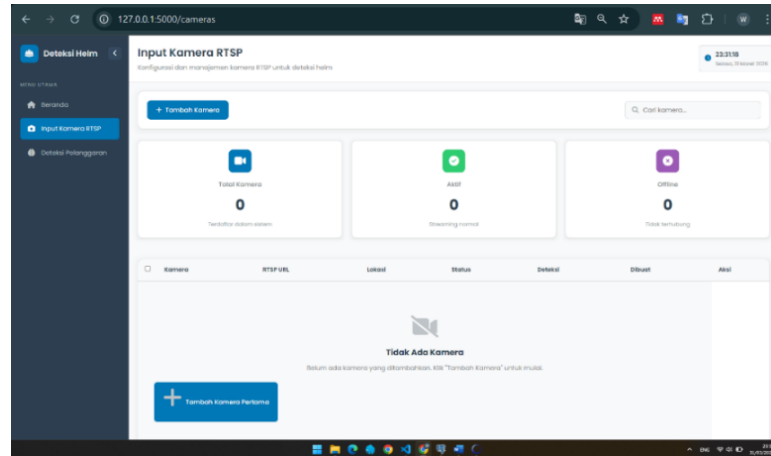
### 3.2 Hasil Implementasi Sistem

Hasil implementasi sistem berupa aplikasi monitoring penggunaan helm keselamatan berbasis web yang terintegrasi dengan model deteksi objek YOLOv11. Sistem dikembangkan menggunakan *framework Flask* dan mampu memproses input *video streaming* secara *real-time* melalui protokol RTSP. Secara umum, sistem bekerja dengan menerima input video dari kamera melalui RTSP, kemudian memproses setiap *frame* menggunakan model YOLOv11 untuk mendeteksi objek pekerja serta mengklasifikasikan penggunaan helm. Hasil deteksi ditampilkan pada *dashboard* dalam bentuk *bounding box* dan label klasifikasi, serta disimpan sebagai data pelanggaran apabila terdeteksi pekerja yang tidak menggunakan helm.



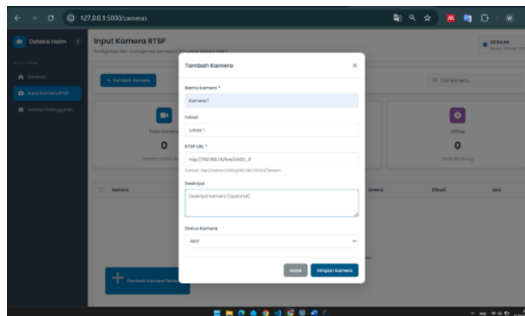
Gambar 13. Dashboard

Gambar 13 menunjukkan tampilan awal *dashboard* monitoring setelah sistem berhasil dijalankan. Pada tahap ini, antarmuka sistem telah berhasil dimuat dengan baik, namun belum menampilkan *video streaming* karena belum terdapat input kamera yang terhubung. Tampilan *dashboard* menunjukkan struktur informasi yang akan digunakan dalam proses monitoring, seperti area tampilan video dan indikator data deteksi. Hal ini menunjukkan bahwa sistem *backend* dan *frontend* telah terintegrasi dengan baik dan siap untuk menerima input data.



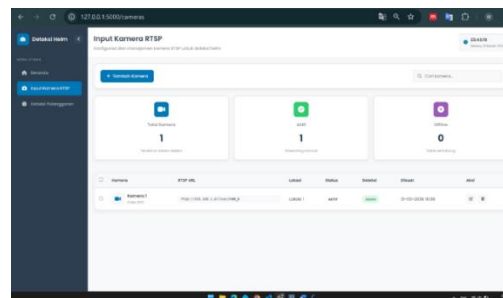
**Gambar 14.** Halaman Data Kamera

Gambar 14 menunjukkan proses pengelolaan sumber video melalui input kamera berbasis RTSP. Sistem memungkinkan pengguna untuk menambahkan dan mengelola data kamera sebagai sumber input dalam proses deteksi objek.



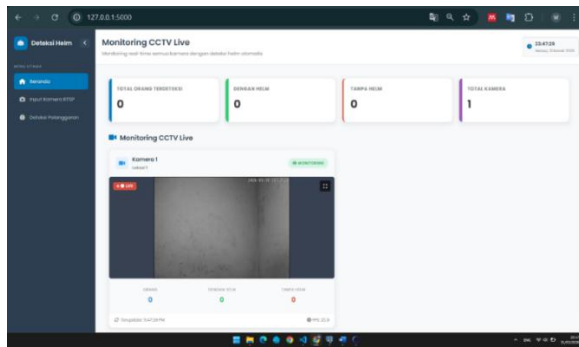
**Gambar 15.** Proses *Input* Kamera

Gambar 15 menunjukkan proses penambahan data kamera ke dalam sistem melalui fitur manajemen kamera. Pada tahap ini, pengguna memasukkan informasi kamera seperti alamat RTSP dan data pendukung lainnya melalui form yang tersedia. Fitur ini berfungsi untuk mendefinisikan sumber video yang akan digunakan dalam proses deteksi.

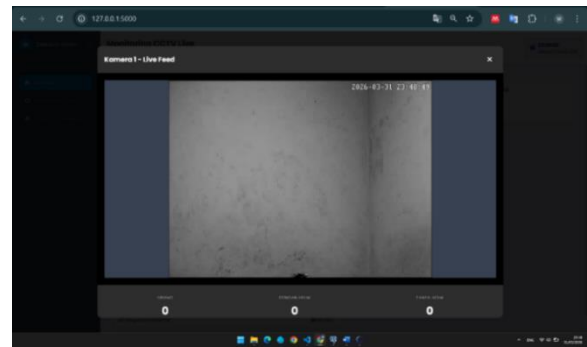


**Gambar 16.** Data Kamera Berhasil Ditambahkan

Gambar 16 menunjukkan hasil setelah data kamera berhasil ditambahkan. Data kamera ditampilkan dalam bentuk tabel sebagai bagian dari manajemen sumber video. Pada tahap ini, sistem telah mampu menerima, menyimpan, dan menampilkan data kamera, namun belum melakukan proses *streaming* maupun deteksi objek.



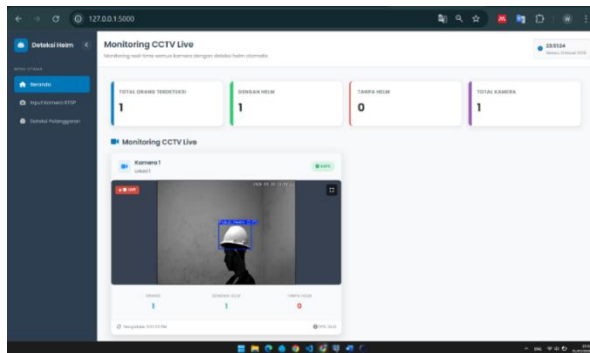
(a) Tampilan *Streaming Kamera*



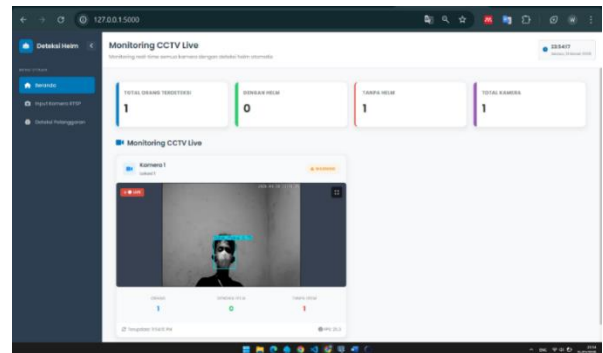
(b) Tampilan *Fullscreen Streaming*

**Gambar 17.** Tampilan: *Streaming Kamera* dan (b) *Fullscreen Streaming*

Gambar 17(a) dan Gambar 17(b) menunjukkan tampilan *video streaming* yang diperoleh dari sumber kamera melalui protokol RTSP. Pada tahap ini, sistem berhasil mengambil dan memproses aliran video dari alamat RTSP yang telah dikonfigurasi, kemudian menampilkannya pada *dashboard* secara kontinu. Hasil ini menunjukkan bahwa integrasi antara sumber kamera dan sistem berjalan dengan baik, serta membuktikan bahwa protokol RTSP dapat diproses dengan benar sebagai input video dalam sistem.



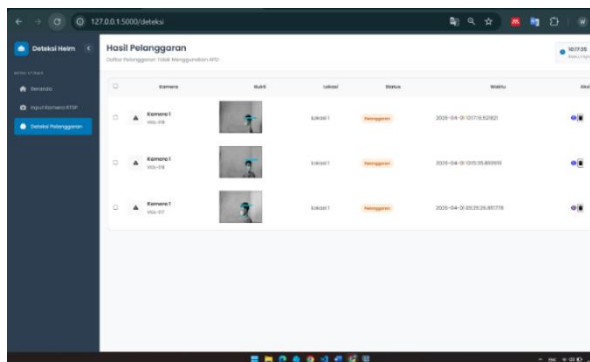
(a) Objek Pakai Helm



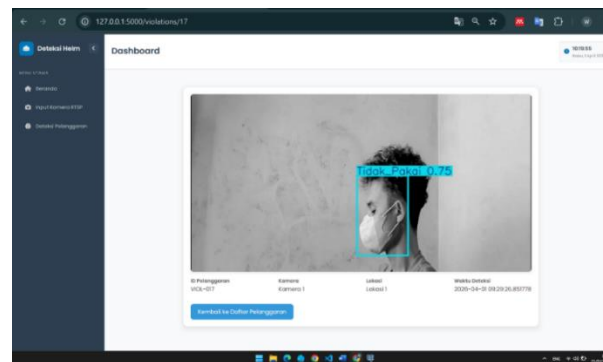
(b) Objek Tanpa Helm

**Gambar 18.** Hasil Deteksi: (a) Objek Pakai Helm dan Objek Tanpa Helm

Gambar 18(a) dan Gambar 18(b) menunjukkan hasil deteksi objek pada *video streaming* yang telah diproses menggunakan model YOLOv11. Sistem berhasil mendeteksi objek pekerja dan memberikan *bounding box* serta label klasifikasi Pakai\_Helm dan Tidak\_Pakai secara *real-time*. Hasil ini menunjukkan bahwa model deteksi telah terintegrasi dengan sistem *streaming* dan mampu melakukan deteksi objek secara kontinu pada setiap *frame video*.



(a) Data Pelanggaran ke *Database*



(b) Data Pelanggaran pada Sistem

**Gambar 19.** Penyimpanan Data Pelanggaran: (a) Pelanggaran ke *Database* dan Pelanggaran pada Sistem

Gambar 19(a) dan Gambar 19(b) menunjukkan data pelanggaran yang dihasilkan dari proses deteksi objek dan telah berhasil disimpan ke dalam *database*. Setiap entri merepresentasikan kejadian pekerja yang tidak menggunakan helm yang terdeteksi secara otomatis oleh sistem. Hasil ini menunjukkan bahwa sistem tidak hanya mampu melakukan deteksi objek, tetapi juga berhasil mengintegrasikan hasil deteksi ke dalam proses penyimpanan data secara otomatis.

### 3.3 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi berjalan sesuai dengan yang diharapkan. Metode yang digunakan adalah *black-box* testing, yaitu pengujian yang mengevaluasi kesesuaian fungsi sistem berdasarkan hubungan antara input dan output tanpa mempertimbangkan struktur internal program. Pengujian difokuskan pada beberapa aspek utama, yaitu fungsionalitas input kamera, proses deteksi objek, penyimpanan data pelanggaran, serta tampilan antarmuka sistem. Hasil pengujian fungsional sistem ditunjukkan pada Tabel 4, yang menunjukkan bahwa seluruh fungsi utama sistem telah berjalan sesuai dengan kebutuhan yang telah dirancang.

**Tabel 4.** Pengujian Fungsionalitas

Skenario Pengujian	Input	Output yang Diharapkan	Output Aktual
Input RTSP Kamera	URL RTSP	Video streaming tampil	Valid
Deteksi Pakai Helm	Video pakai helm	Label Pakai_Helm muncul	Valid
Deteksi Tidak Pakai Helm	Video tanpa helm	Label Tidak_Pakai muncul	Valid
Penyimpanan Pelanggaran	Deteksi pelanggaran	Data masuk ke database	Valid
Tampilan Dashboard	Akses halaman	Video & statistik tampil	Valid
Stabilitas Sistem	Streaming terus-menerus	Sistem tetap berjalan tanpa error	Valid

### 3.4 Pembahasan

Berdasarkan hasil pelatihan dan evaluasi, model YOLOv11 menunjukkan performa deteksi yang tinggi dengan mAP@0.5 sebesar 0.905. Hasil ini menunjukkan kemampuan model dalam mengenali objek dengan akurasi yang baik, terutama pada data yang mirip dengan data pelatihan. Stabilitas penurunan loss serta kesesuaian kurva training dan validation mengindikasikan kemampuan generalisasi yang cukup baik, meskipun masih terbatas pada distribusi data penelitian. Namun demikian, performa model belum sepenuhnya konsisten pada seluruh kelas. Recall yang lebih rendah pada kelas Tidak\_Pakai menunjukkan adanya kegagalan deteksi pada kondisi tertentu. Hal ini menunjukkan bahwa model lebih sensitif terhadap objek dengan ciri visual yang jelas seperti helm, dibandingkan objek tanpa atribut yang lebih ambigu sehingga model belum sepenuhnya robust terhadap variasi kondisi nyata.

Hasil *confusion matrix* menunjukkan masih adanya kesalahan deteksi pada area *background* yang mengindikasikan keterbatasan model dalam membedakan objek dan non-objek pada kondisi visual yang kompleks. Permasalahan ini umum terjadi pada model *deep learning* yang bergantung pada kualitas dan keragaman dataset. Sistem yang dikembangkan telah berjalan secara *real-time* dan siap diterapkan pada skenario monitoring, meskipun performanya masih dipengaruhi oleh kualitas *input* video, seperti pencahayaan, sudut pandang, dan resolusi kamera. Oleh karena itu, peningkatan *robustness* melalui penambahan variasi dataset dan optimalisasi model terhadap kondisi lingkungan yang lebih beragam masih diperlukan.

Hasil penelitian ini menunjukkan peningkatan dibandingkan penelitian terdahulu. Penelitian Fatkhin dan Fadjeri [5] yang menggunakan YOLOv8 memperoleh *precision* sebesar 77,7%, *recall* 60,7%, dan mAP sebesar 68,9%, sedangkan pada penelitian ini YOLOv11 memperoleh *precision* sebesar 0,881, *recall* sebesar 0,865, dan mAP@0.5 sebesar 0,905. Hasil tersebut menunjukkan bahwa YOLOv11 memiliki kemampuan deteksi yang lebih baik dalam mengenali penggunaan helm keselamatan pada kondisi *real-time*. Selain itu, penelitian Iman et al. [11] menggunakan YOLOv8 mampu melakukan deteksi *real-time* dengan tingkat kepercayaan di atas 90%, namun belum mengintegrasikan sistem monitoring berbasis web secara menyeluruh. Pada penelitian ini, YOLOv11 telah diintegrasikan dengan sistem monitoring berbasis web serta *video streaming* RTSP secara *real-time*, sehingga lebih mendukung implementasi pada lingkungan konstruksi.

## 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi helm keselamatan kerja berbasis *computer vision* berhasil dikembangkan menggunakan metode *Rapid Application Development* (RAD) dan algoritma YOLOv11. Model yang dilatih mampu mencapai performa yang baik dengan nilai *precision* sebesar 0,88, *recall* sebesar 0,86, serta mAP@0.5 sebesar 0,90, yang menunjukkan kemampuan model dalam mendeteksi objek secara akurat. Hasil evaluasi menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik terhadap data baru, meskipun masih terdapat kesalahan deteksi terutama pada kondisi tertentu seperti pencahayaan yang kurang optimal dan sudut pengambilan gambar yang tidak ideal. Selain itu, perbedaan performa antar kelas menunjukkan bahwa deteksi pada kelas Tidak\_Pakai masih memiliki tantangan dalam mengurangi kesalahan berupa *false negative*. Oleh karena itu, untuk pengembangan selanjutnya disarankan dilakukan penyeimbangan rasio dataset antar kelas serta penerapan teknik augmentasi yang lebih spesifik pada skenario objek tanpa helm guna meningkatkan kemampuan model dalam mengenali variasi kondisi visual pada kelas Tidak\_Pakai. Implementasi model ke dalam sistem berbasis web memungkinkan proses monitoring dilakukan secara *real-time* melalui integrasi dengan kamera berbasis RTSP. Hasil pengujian sistem menunjukkan

bahwa seluruh fungsi utama, seperti *video streaming*, deteksi objek, dan pencatatan pelanggaran, berjalan dengan baik sesuai dengan kebutuhan. Dengan demikian, sistem yang dikembangkan dapat mendukung pengawasan penggunaan helm keselamatan kerja di lingkungan industri. Peningkatan performa sistem pada berbagai kondisi lingkungan masih diperlukan agar deteksi dapat berjalan lebih optimal.

## REFERENCES

- [1] N. Rahmawati and K. Puji, "Pengaruh Program K3 (Keselamatan dan Kesehatan Kerja) terhadap Produktivitas Kerja Karyawan pada PT. Telkom Witel Bandar Lampung," *J. Ekon. Ef.*, vol. 7, no. 1, pp. 131–137, 2024, doi: 10.32493/JEE.v7i1.44420.
- [2] E. Haryanto *et al.*, *Kesehatan dan Keselamatan Kerja Tenaga Kerja*. Purbalingga: Eureka Media Aksara, 2024.
- [3] M. E. Laily, F. N. Fajri, and G. Q. O. Pratamasunu, "Deteksi Penggunaan Alat Pelindung Diri (APD) Untuk Keselamatan dan Kesehatan Kerja Menggunakan Metode Mask Region Convolutional Neural Network (Mask R-CNN)," *J. Komput. Terap.*, vol. 8, no. 2, pp. 279–288, 2022, doi: 10.35143/jkt.v8i2.5732.
- [4] F. Edigan, L. R. P. Sari, and R. Amalia, "Hubungan Antara Perilaku Perilaku Keselamatan Kerja Terhadap Penggunaan Alat Pelindung Diri (APD) Pada Karyawan PT Surya Agrolika Reksa Di Sei. Basau," *J. Sainis*, vol. 19, no. 02, p. 61, 2019, doi: 10.25299/saintis.2019.vol19(02).3741.
- [5] N. Fatkhin and A. Fadjeri, "Pembelajaran Mesin Untuk Deteksi Helm Keselamatan Menggunakan Algoritma YOLOv8," *J. Ilm. SINUS*, vol. 22, no. 2, p. 77, 2024, doi: 10.30646/sinus.v22i2.843.
- [6] D. N. Putri and F. Lestari, "Analisis penyebab kecelakaan kerja pada pekerja di proyek konstruksi : Literatur review," *J. Kesehat. Masy.*, vol. 7, no. 1, pp. 451–452, 2023, doi: 10.31004/prepotif.v7i1.13281.
- [7] H. Supriyanto, S. C. Abadi, and A. Shalsabilah, "Deteksi Helm Keselamatan Menggunakan Jetson Nano dan YOLOv7," *J. Appl. Comput. Sci. Technol.*, vol. 5, no. 1, pp. 1–8, 2024, doi: 10.52158/jacost.v5i1.637.
- [8] F. Fersellia, A. Lutfiyani, F. Fachri, and E. Wahyuningsih, "Deteksi Hewan Secara Real-Time Menggunakan Algoritma You Only Look Once (YOLO)," *INSOLOGI J. Sains dan Teknol.*, vol. 5, no. 1, pp. 263–269, Feb. 2026, doi: 10.55123/insologi.v5i1.7592.
- [9] I. A. Zulkarnain and Kusriani, "Optimasi Yolov11 Melalui Hyperparameter Tuning dan Data Augmentasi untuk Meningkatkan Akurasi Deteksi Kendaraan pada Kondisi Malam Hari," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 5, no. October, pp. 1294–1303, 2025, doi: 10.57152/malcom.v5i4.2250.
- [10] B. Widodo, H. Armanto, and E. Setyati, "Deteksi Pemakaian Helm Proyek Dengan Metode Convolutional Neural Network," *J. Intell. Syst. Comput.*, vol. 3, no. 1, pp. 23–29, Apr. 2021, doi: 10.52985/insyst.v3i1.157.
- [11] R. M. Iman, Srimuryati, and R. Rahmadewi, "Deteksi Penggunaan Helm Safety Pada Pekerja Menggunakan Algoritma You Only Look Once Versi 8 (YOLOv8)," *TESLA J. Tek. Elektro*, vol. 27, no. 1, pp. 11–18, Apr. 2025, doi: 10.24912/tesla.v27i1.33391.
- [12] H. A. Reswara, B. Priyatna, A. Hananto, and T. Tukino, "Implementasi Deteksi Objek Penggunaan Helm Dengan Metode YOLOv10," *J. Minfo Polgan*, vol. 14, no. 1, pp. 1380–1387, 2025, doi: 10.33395/jmp.v14i1.15010.
- [13] N. A. K. D. Pasongko, A. Khairunnisa, and S. Aras, "Deteksi Penggunaan Safety Helmet Menggunakan YOLOv5," *J. Inf. Eng. Educ. Technol.*, vol. 7, no. 2, pp. 74–77, Dec. 2023, doi: 10.26740/jieet.v7n2.p74-77.
- [14] R. Abdillah, R. Hermawan, A. Kuncoro, A. Adi Trinoto, F. Hizryan, and H. Arifin, "Analisis dan Implementasi Sistem Informasi Persewaan Alat dengan metode Rapid Application Development," *J. Penelit. Sist. Inf.*, vol. 1, no. 3, pp. 23–37, Jul. 2023, doi: 10.54066/jpsi.v1i3.633.
- [15] Z. Niqotaini *et al.*, *Rekayasa Perangkat Lunak*. Yogyakarta: PT Penamuda Media, 2023.
- [16] R. N. Firmansyah, Y. R. Ramadhan, and C. D. Lestari, "Rancang Bangun Aplikasi Pengajuan Cuti Karyawan Berbasis Web pada PT. Indorama Synthetics," *J. Ris. Sist. Inf.*, vol. 2, no. 3, pp. 99–109, Jul. 2025, doi: 10.69714/70gd7651.
- [17] A. Hafisah, A. A. B. Kaban, S. L. Ramadhan, and Nurbaiti, "Keamanan Data Dalam Sistem Database," *J. Ilm. Nusant.*, vol. 2, no. 4, pp. 183–197, 2025, doi: 10.61722/jinu.v2i4.4997.
- [18] H. L. Walingkas and P. O. N. Saian, "Penerapan Framework Flask pada Pembangunan Sistem Informasi Pemasok Barang," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 7, no. 2, 2023, doi: 10.35870/jtik.v7i2.729.
- [19] Roboflow Universe Projects, "Hard Hats Computer Vision Model," Roboflow Universe. Accessed: Jun. 04, 2026. [Online]. Available: <https://universe.roboflow.com/roboflow-universe-projects/hard-hats-fhbh5>
- [20] M. H. Achmad, A. Pramudwiatmoko, M. S. Gumilang, B. Al Karim, and H. Wiyono, "Analisis Kinerja Model Deteksi Objek YOLO, SSD, dan Faster R-CNN pada Citra Penglihatan Malam untuk Pengenalan Tindak Kejahatan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 12, no. 1, pp. 145–152, Feb. 2025, doi: 10.25126/jtiik.2025128409.
- [21] R. N. Singarimbun, E. B. Nababan, and O. S. Sitompul, "Adaptive Moment Estimation to Minimize Square Error In Backpropagation Algorithm," *J. Comput. Appl. Informatics*, vol. 04, no. 1, pp. 27–46, 2020, doi: 10.32734/jocai.v4.i1-1160.
- [22] W. R. N. Rifa'i, D. Erwanto, and I. Yanuartanti, "Evaluasi Kinerja CNN dengan Optimizer RMSprop, Adam dan SGD dalam Klasifikasi Penyakit Daun Anggur," *J. Tek. Elektro Electron. Control. Telecommun. Comput. Inf. Power Syst.*, pp. 91–104, 2025, doi: 10.30736/je-unisla.v10i1.1409.
- [23] C. Ciksadan, S. Soim, and N. Jami, "Desain dan Pengembangan Website untuk Mendeteksi Malware Menggunakan Framework Flask yang Diintegrasikan dengan Machine Learning," *J. Teknol. Sist. Inf. dan Apl.*, vol. 7, no. 3, pp. 1213–1218, 2024, doi: 10.32493/jtsi.v7i3.42003.
- [24] E. Suherlan, S. Arti, S. Nabilah, and Z. Hazimah, "Penerapan Flask Framework Untuk Deployment Model Machine Learning Dalam Mendukung Analisis Adaptasi Mahasiswa Pada Pembelajaran Daring," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 9, no. 1, pp. 110–117, Jun. 2025, doi: 10.21009/pinter.9.1.15.