

***PROTOTYPE DETEKTOR PERLENGKAPAN KESEHATAN DAN
KESELAMATAN KERJA (K3) BERBASIS DEEP LEARNING YOLO***

Tesis S-2

Untuk memenuhi persyaratan memperoleh gelar Magister Teknik
Program Studi Magister Teknik Elektro



Diajukan oleh :

Viky Ananta Sutrisno

20602300052

PROGRAM STUDI MAGISTER TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2024

PROTOTYPE DETEKTOR PERLENGKAPAN KESEHATAN DAN KESELAMATAN KERJA (K3) BERBASIS DEEP LEARNING YOLO

Tesis S-2

Untuk memenuhi persyaratan memperoleh gelar Magister Teknik
Program Studi Magister Teknik Elektro



Diajukan oleh :

Viky Ananta Sutrisno

20602300052

PROGRAM STUDI MAGISTER TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2024

HALAMAN PENGESAHAN PEMBIMBING

Tesis S-2

PROTOTYPE DETEKTOR PERLENGKAPAN KESEHATAN DAN KESELAMATAN KERJA (K3) BERBASIS DEEP LEARNING YOLO

yang diajukan oleh:

Viky Ananta Sutrisno

20602300052

Telah diperiksa dan disetujui,
Pada tanggal 06 Desember 2024

Pembimbing

Dr. Eka Nuryanto Budisusila, S.T., M.T.

NIDN: 0619107301

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Teknik
Tanggal 06 Desember 2024

Ketua Program Studi Magister Teknik Elektro

Prof. Dr. Ir. Sri Astuti Dwi Prasetyowati, M.Si.

NIDN: 0620026501

HALAMAN PENGESAHAN

**PROTOTYPE DETEKTOR PERLENGKAPAN KESEHATAN DAN
KESELAMATAN KERJA (K3) BERBASIS DEEP LEARNING YOLO**

yang dipersiapkan dan disusun oleh

Viky Ananta Sutrisno

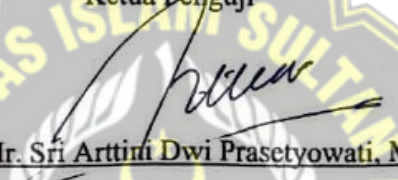
20602300052

telah dipertahankan di depan Dewan Penguji

Pada tanggal 06 Desember 2024

Susunan Dewan Penguji


Ketua Penguji



Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si.

NIDN: 0620026501


Penguji I



Dr. Eka Nuryanto Budisusila, S.T., M.T.

NIDN: 0619107301

Penguji II



Dr. Ir. Sukarno Budi Utomo, M.T.

NIDN: 0619076401

Tesis ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Magister Teknik

Tanggal 06 Desember 2024

Ketua Program Studi Magister Teknik Elektro



Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si.

NIDN: 0620026501

PERNYATAAN KEASLIAN TESIS

Saya yang bertanda tangan dibawah ini:

Nama : Viky Ananta Sutrisno

NIM : 20602300052

Program Studi : Magister Teknik Elektro

Fakultas : Teknologi Industri

Dengan ini saya menyatakan bahwa Tesis yang diajukan kepada Program Studi Magister Teknik Elektro dengan Judul:

“*Prototype* Detektor Perlengkapan Kesehatan Dan Keselamatan Kerja (K3) Berbasis *Deep Learning* Yolo”

Adalah hasil karya sendiri, judul tersebut belum pernah diajukan untuk memperoleh gelar Magister Teknik pada Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung (UNISSULA) ataupun pada universitas lain serta belum pernah ditulis maupun diterbitkan oleh orang lain kecuali secara tertulis diacu, disitasi dan ditunjuk dalam daftar pustaka. Tesis ini adalah milik saya, segala bentuk kesalahan dan kekeliruan dalam Tesis ini adalah tanggung jawab saya.

Semarang, 06 Desember 2024



Viky Ananta Sutrisno

20602300052

PRAKATA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tesis yang berjudul "*Prototype Detektor Perlengkapan Kesehatan dan Keselamatan Kerja (K3) Berbasis Deep Learning YOLO*". Tesis ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Magister Teknik Elektro di Universitas Islam Sultan Agung Semarang. Penulisan tesis ini tentunya tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Dengan demikian, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Dr. Eka Nuryanto Budisusila, S.T., M.T., selaku pembimbing utama, atas segala arahan, bimbingan, dan dukungan yang diberikan selama proses penyusunan tesis ini.
2. Dosen dan staf pengajar di Program Magister Teknik Elektro Universitas Islam Sultan Agung Semarang yang telah memberikan ilmu dan inspirasi yang sangat berharga selama masa studi.
3. Keluarga tercinta yang senantiasa memberikan dukungan moral dan material, serta doa yang tiada henti.
4. Teman-teman dan rekan mahasiswa yang selalu memberikan semangat dan kerja sama selama masa studi.

Penulis menyadari bahwa tesis ini masih memiliki kekurangan dan penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan tesis ini di masa mendatang. Semoga tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan, khususnya dalam bidang teknologi deteksi keselamatan kerja berbasis deep learning, serta menjadi kontribusi yang berguna bagi masyarakat dan industri.

Semarang, 06 Desember 2024



Viky Ananta Sutrisno

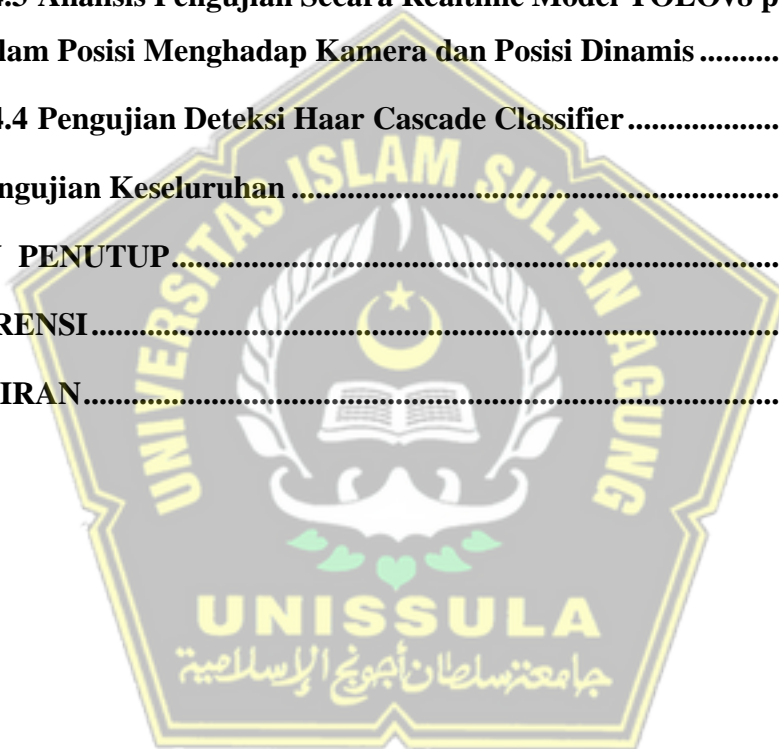
DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN PEMBIMBING.....	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TESIS	v
PRAKATA.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
ABSTRAK	xiv
ABSTRACT.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah	4
1.5 Keaslian Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	8
2.1 Tinjauan Pustaka	8
2.2 Landasan Teori	10
2.2.1 Kesehatan dan Keselamatan Kerja (K3).....	10
2.2.2 Citra	11
2.2.2.1 Citra Digital.....	11
2.2.3 <i>Deep Learning</i>	14
2.2.4 <i>Convolutional Neural Network</i> (CNN).....	14

2.2.5 YOLO (You Only Look Once)	18
2.2.5.1 Desain Jaringan YOLO	19
2.2.6 Haar Cascade Classifier	25
2.2.6.1 Haar feature.....	25
2.2.6.2 <i>Integral image</i>	27
2.2.6.3 Adaboost (<i>Adaptive Boosting</i>).....	28
2.2.6.4 Cascade Classifier	29
2.2.7 Roboflow	29
2.2.8 Google Colaboratory	31
BAB III METODE PENELITIAN	32
3.1 Model Penelitian.....	32
3.2 Diagram Alir.....	32
3.3 Alat Dan Bahan Penelitian	33
3.4 Desain Perancangan System	33
3.5 Sistem <i>processing</i> Data	36
3.5.1 Akuisisi Data	38
3.5.2 Preprocessing Data.....	38
3.6 Training	39
3.7 Arsitektur Jaringan Yolo	40
3.6.1 Pendeteksian dengan Yolo	40
3.8 Algoritma Haarcascade Classifier	42
3.9 Validasi.....	43
3.10 Pengambilan Data dan Pengujian Data	45
3.10.1 Pengujian Deteksi Offline	45
3.10.2 Pengujian Deteksi <i>Online/Realtime</i>	45
3.10.3 Pengujian Keseluruhan.....	45

3.11 Lokasi Penelitian	45
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	46
4.1 Preparation Dataset	46
4.1.1 Akuisisi Data	46
4.1.2 Anotasi Data	47
4.2 Training dan Validasi	48
4.2.1 Model YOLOV8.....	48
4.2.2 Nilai Batch.....	53
4.2.3 Nilai Epoch	54
4.3 Pengujian Secara Offline.....	55
4.3.1 Pengujian Deteksi Offline Model YoloV8 Menghadap Kamera	55
4.3.1.1 Pengujian Deteksi Offline Menghadap Kamera Dengan Variasi Jarak Untuk Helm	55
4.3.1.2 Pengujian Deteksi Offline Menghadap Kamera Dengan Variasi Jarak Untuk Vest.....	59
4.3.2 Pengujian Deteksi Offline Model YoloV8 Pada Posisi Dinamis.....	62
4.3.2.1 Pengujian Deteksi Offline Pada Posisi Dinamis Dengan Variasi Jarak Untuk Helm	63
4.3.2.2 Pengujian Deteksi Offline Pada Posisi Dinamis Dengan Variasi Jarak Untuk Vest.....	66
4.3.3 Analisis Pengujian Secara Offline Model YOLOv8 pada Objek dalam Posisi Menghadap Kamera dan Posisi Dinamis	70
4.4 Pengujian Secara Online atau Realtime	72
4.4.1 Pengujian Deteksi Secara Realtime Model YoloV8 Menghadap Kamera	72
4.4.1.1 Pengujian Deteksi Secara Real-Time Saat Menghadap Kamera Dengan Variasi Jarak Untuk Helm.....	73

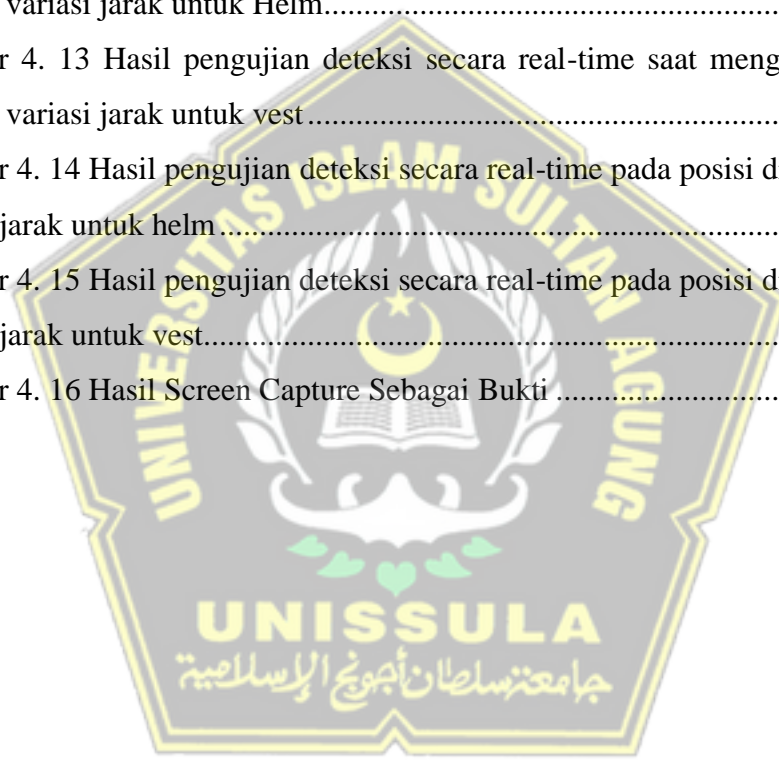
4.4.1.2 Pengujian Deteksi Secara Real-Time Saat Menghadap Kamera Dengan Variasi Jarak Untuk Vest	76
4.4.2 Pengujian Deteksi Secara Realtime Model YOLOV8 Posisi Dinamis	79
4.4.2.1 Pengujian Deteksi Secara Real-Time Pada Posisi Dinamis Dengan Variasi Jarak Untuk Helm.....	79
4.4.2.2 Pengujian Deteksi Secara Real-Time Pada Posisi Dinamis Dengan Variasi Jarak Untuk Vest.....	83
4.4.3 Analisis Pengujian Secara Realtime Model YOLOv8 pada Objek dalam Posisi Menghadap Kamera dan Posisi Dinamis	86
4.4.4 Pengujian Deteksi Haar Cascade Classifier	87
4.5 Pengujian Keseluruhan	90
BAB V PENUTUP.....	93
REFERENSI.....	94
LAMPIRAN.....	97



DAFTAR GAMBAR

Gambar 2. 1 Koordinat Citra Digital	12
Gambar 2. 2 Representasi	13
Gambar 2. 3 Perbedaan dimensi MLP dengan CNN	15
Gambar 2. 4 Contoh arsitektur CNN.....	16
Gambar 2. 5 Lapisan pada CNN	16
Gambar 2. 6 Arsitektur dasar YOLO	20
Gambar 2. 7 Sistem deteksi YOLO.....	21
Gambar 2. 8 Bounding Box pada YOLO.....	22
Gambar 2. 9 Model YOLO	22
Gambar 2. 10 Jenis haar feature.....	25
Gambar 2. 11 Perbedaan citra asil dan grayscale.....	27
Gambar 2. 12 Pixel tetangga pada proses integral image	28
Gambar 2. 13 Roboflow	30
Gambar 2. 14 Alur kerja platform roboflow	30
Gambar 3. 1 Diagram alir penelitian.....	32
Gambar 3. 2 Diagram Alir kerja sistem	33
Gambar 3. 3 Diagram sistem keseluruhan	34
Gambar 3. 4 Diagram blok sistem processing data.....	37
Gambar 3. 5 Akuisisi data gambar.....	38
Gambar 3. 6 Anotasi menggunakan roboflow	39
Gambar 3. 7 Diagram alir pendeteksian YOLO.....	41
Gambar 3. 8 Alur Haar Cascade Clasifier.....	43
Gambar 4. 1 Gambar objek manusia menggunakan helm	46
Gambar 4. 2 Gambar objek manusia menggunakan helm dan rompi keselamatan	47
Gambar 4. 3 Proses anotasi data dengan website roboflow	47
Gambar 4. 4 Persebaran dataset	48
Gambar 4. 5 Training di Google Colaboratory	48
Gambar 4. 6 Hasil Training dengan Arsitektur Yolov8.....	49
Gambar 4. 7 Confusion matrix YOLOv8.....	50

Gambar 4. 8 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk helm.....	58
Gambar 4. 9 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk vest	61
Gambar 4. 10 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk helm.....	65
Gambar 4. 11 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk vest	69
Gambar 4. 12 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk Helm.....	75
Gambar 4. 13 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk vest.....	78
Gambar 4. 14 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk helm	81
Gambar 4. 15 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk vest.....	85
Gambar 4. 16 Hasil Screen Capture Sebagai Bukti	91



DAFTAR TABEL

Tabel 1.1 Perbedaan dengan penelitian-penelitian sebelumnya.....	4
Tabel 3. 1 Tabel Confusion Matrix	44
Tabel 4. 1 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk helm.....	56
Tabel 4. 2 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk vest	59
Tabel 4. 3 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk helm.....	63
Tabel 4. 4 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk vest	67
Tabel 4. 5 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk helm.....	73
Tabel 4. 6 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk vest.....	76
Tabel 4. 7 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk helm	79
Tabel 4. 8 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk vest.....	83
Tabel 4. 9 Hasil pengujian model Haar cascade dengan variasi jarak	87
Tabel 4. 10 Hasil Pengujian Keseluruhan.....	90

ABSTRAK

PROTOTYPE DETEKTOR PERLENGKAPAN KESEHATAN DAN KESELAMATAN KERJA (K3) BERBASIS DEEP LEARNING YOLO; Viky Ananta Sutrisno, 20602300052; Jurusan Teknik Elektro, Universitas Islam Sultan Agung, Semarang, Indonesia.

Menurut International Labour Organization (ILO), setiap tahun terjadi lebih dari 250 juta kecelakaan kerja dan lebih dari 160 juta pekerja menderita sakit akibat mengalami kecelakaan di tempat kerja dan terdapat lebih dari 1,2 juta pekerja kehilangan nyawa akibat kecelakaan dan kesehatan terkait pekerjaan. Tingginya nilai kecelakaan kerja membuat PT. Prima Barra Sejahtera sebagai perusahaan yang bergerak di bidang jasa general kontraktor, membutuhkan sistem pengawasan K3 yang memadai untuk memantau ketertiban (APD) yang wajib digunakan oleh karyawan.

Sistem pendeteksian objek berupa helm dan rompi keselamatan dengan metode Convolutional Neural Network algoritma yolov8. Sebelum implementasi, disiapkan dataset sebanyak 5.714 data yang telah dianotasi menggunakan platform Roboflow untuk keperluan pelatihan model. Proses pelatihan dilakukan melalui Google Colaboratory dengan pengaturan hyperparameter berupa 25 epoch dan batch size sebesar 16. Hasil dari proses pelatihan ini menghasilkan model dengan nilai mean Average Precision (mAP) sebesar 0.67305.

Pengujian dilakukan pada lokasi kerja menggunakan kamera webcam Brio 100 1080/30 FPS yang dioperasikan melalui platform open-source Visual Studio Code (VS Code). Pelanggaran yang terdeteksi dievaluasi menggunakan mean Average Precision (mAP). Sistem ini secara otomatis mengklasifikasikan pelanggaran dan menghasilkan bukti dalam bentuk tangkapan layar. Pengujian terbaik diperoleh pada jarak 3 meter, dengan akurasi 90% untuk deteksi rompi dan 89% untuk deteksi helm di 3 FPS.

Kata kunci: Kesehatan Dan Keselamatan Kerja, Deep Learning, YOLO.

ABSTRACT

DEEP LEARNING YOLO-BASED HEALTH AND SAFETY EQUIPMENT DETECTOR PROTOTYPE; Viky Ananta Sutrisno, 20602300052; Department of Electrical Engineering, Islam Sultan Agung University, Semarang, Indonesia

According to the International Labour Organization (ILO), every year, more than 250 million workplace accidents occur, and over 160 million workers suffer illnesses caused by workplace incidents. Additionally, more than 1.2 million workers lose their lives due to work-related accidents and health issues. The high number of workplace accidents has prompted PT. Prima Barra Sejahtera, a company engaged in general contracting services, to require an adequate occupational health and safety (OHS) monitoring system to ensure compliance with the use of personal protective equipment (PPE) by employees.

The object detection system for helmets and safety vests utilizes the YOLOv8 algorithm based on Convolutional Neural Networks. Before implementation, a dataset consisting of 5,714 annotated data points was prepared using the Roboflow platform for model training purposes. The training process was conducted on Google Colaboratory with hyperparameters set to 25 epochs and a batch size of 16. This training process produced a model with a mean Average Precision (mAP) of 0.67305.

Testing was conducted at a worksite using a Brio 100 1080/30 FPS webcam operated through the open-source platform Visual Studio Code (VS Code). Detected violations were evaluated using mean Average Precision (mAP). The system automatically classifies violations and generates evidence in the form of screenshots. The best testing results were achieved at a distance of 3 meters, with 90% accuracy for vest detection and 89% accuracy for helmet detection at 3 FPS.

Keyword: Occupational Health and Safety, Deep Learning, YOLO

BAB I PENDAHULUAN

1.1 Latar Belakang

Kesehatan dan keselamatan Kerja merupakan masalah paling vital untuk tercapainya tujuan dalam suatu proyek pada bidang konstruksi. Menurut International Labour Organization (ILO), setiap tahun terjadi lebih dari 250 juta kecelakaan kerja dan lebih dari 160 juta pekerja menderita sakit akibat mengalami kecelakaan di tempat kerja. Setidaknya terdapat lebih dari 1,2 juta pekerja kehilangan nyawa akibat kecelakaan dan kesehatan terkait pekerjaan [1]. Tingginya angka kecelakaan kerja ini disinyalir bersumber dari kurangnya kesadaran pekerja dalam mematuhi peraturan keselamatan kerja, sebagaimana diatur dalam Peraturan Menteri Tenaga Kerja dan Transmigrasi No.Per.08/Men/VII/2010 tentang Alat Pelindung Diri (APD).

Pelanggaran K3 ini masih marak terjadi di sejumlah proyek konstruksi, dipicu oleh lemahnya pengawasan oleh tim K3. Berkaitan dengan sistem pemantauan dan deteksi peralatan K3 karyawan, banyak perusahaan masih melakukannya secara manual, termasuk di PT. Prima Barra Sejahtera (PBS). PT. PBS masih menggunakan sistem *monitoring* dan deteksi perlengkapan K3 karyawan secara manual. Hal ini dinilai kurang efektif karena keterlibatan manusia dalam melakukan pengecekan dapat mengakibatkan inefisiensi dan durasi pengecekan yang membutuhkan waktu lama. PT. PBS sebagai perusahaan yang bergerak di bidang jasa general kontraktor, membutuhkan sistem pengawasan K3 yang memadai untuk memantau ketertiban (APD) yang wajib digunakan oleh karyawan. Informasi dari tim K3 PT. PBS bahwa setidaknya beberapa pekerja yang bekerja di *workshop* tidak menggunakan APD yang diwajibkan. Akibatnya, terjadi beberapa insiden terkait tidak menggunakan helm, seperti terbentur kepala pada perancah, terbentur mesin, dan benturan dengan pipa atau struktur baja.

Teknologi dan informasi dalam pengolahan citra dapat menjadi solusi untuk meningkatkan efisiensi dan kecepatan dalam melakukan pengawasan terhadap perlengkapan K3 karyawan. Sistem pengawasan peralatan K3 menggunakan kecerdasan buatan tidak dapat dipisahkan dari proses pendeteksian. Hal ini disebabkan karena pengawasan terhadap perlengkapan K3 merupakan tindakan yang dilakukan setelah perangkat mendeteksi perlengkapan K3 yang digunakan oleh pekerja. Proses pendeteksian berfungsi sebagai masukan kondisi yang akan diproses dan diolah menjadi keputusan oleh perangkat. Pendeteksian perangkat K3 dilakukan dengan bantuan kamera yang dipasang di *workshop* yang memerlukan penggunaan peralatan keselamatan kerja, terutama rompi keselamatan dan helm. Terdapat banyak metode yang dapat digunakan untuk pendeteksian objek, salah satunya adalah metode pendeteksian YOLO (*You Only Look Once*).

Algoritma YOLO menawarkan sejumlah keunggulan signifikan dalam bidang deteksi objek, terutama dalam hal kecepatan pemrosesan dan tingkat akurasi yang tinggi jika dibandingkan dengan metode konvensional lainnya. Kemudahan penggunaan dan pelatihan model YOLO menjadikannya pilihan yang menarik bagi para peneliti pemula. Kemampuan YOLO dalam memproses hingga 45 frame per detik, bahkan mencapai 155 frame per detik dengan versi *fast* YOLO, menunjukkan efisiensi yang luar biasa. Fleksibilitas YOLO telah dibuktikan melalui penerapannya pada beragam proyek deteksi dan pengenalan objek, seperti yang terlihat pada penelitian berjudul *Implementation of the Yolov8 Method to Detect Work Safety* mAP50 mencapai 99,5%. Sedangkan hasil pengujian dengan 100 citra, didapatkan akurasi mencapai 99% [2]. Pada penelitian lain dengan judul Sistem Pendeteksi Alat Pelindung Diri (APD) pada Pekerja Konstruksi Berbasis *Convolutional Neural Network* yang menggunakan kamera dan algoritma deteksi objek YOLOv7 berbasis *Convolutional Neural Network* yang mencapai akurasi yang baik pada jarak ideal 3 meter dan 4,5 meter dengan nilai mAP 0,912 dan 0,947 masing-masing. Pengujian mekanisme alarm menunjukkan akurasi sebesar 1,0 pada jarak 1,5 meter dan 3 meter [3]. Pada penelitian yang serupa dengan judul Implementasi Metode YOLO pada Deteksi Pakaian Keselamatan yang Lengkap di Proyek Kontruksi Sistem pengambilan keputusan menggunakan metode YOLOv5.

140 data gambar yang terdiri dari data 96 train set, 28 valid set dan 16 test set dan berhasil mendeteksi dengan baik dan benar citra menggunakan pakaian keselamatan dan citra tidak menggunakan pakaian keselamatan dengan akurasi 83% [4].

Berdasarkan dari beberapa hasil penelitian tersebut, menunjukkan bahwa YOLO memiliki akurasi dan kecepatan deteksi yang cukup tinggi sehingga sangat cocok digunakan sebagai metode untuk melakukan deteksi objek. Peneliti ini akan berfokus membuat rancangan terkait sistem deteksi perlengkapan K3 berupa Helm dan Rompi untuk mendeteksi penggunaan APD di PT. PBS. Penelitian ini diharapkan dapat mendorong kemajuan teknologi dan berkontribusi dalam upaya meminimalkan kecelakaan kerja yang diakibatkan oleh kurangnya kepedulian pekerja terhadap penggunaan alat keselamatan kerja.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana perancangan sistem detektor perlengkapan kesehatan dan keselamatan kerja (K3) berbasis *deep learning* YOLO.
2. Bagaimana pengaruh jarak objek dalam akurasi sistem detektor perlengkapan kesehatan dan keselamatan kerja (K3) berbasis *deep learning* YOLO.

1.3 Tujuan

Tujuan dari Penelitian ini sebagai berikut:

1. Mengetahui perancangan sistem detektor perlengkapan kesehatan dan keselamatan kerja (K3) berbasis *deep learning* YOLO.
2. Mengetahui pengaruh jarak objek dalam akurasi sistem detektor perlengkapan kesehatan dan keselamatan kerja (K3) berbasis *deep learning* YOLO.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. Metode deteksi menggunakan metode YOLOv8.
2. Penelitian ini berfokus pada pendeteksian penggunaan perlengkapan kerja berupa helm berwarna putih dan oranye serta rompi keselamatan model V.
3. Dataset diambil secara acak dari internet dengan data yang diperoleh secara langsung.
4. Mengabaikan kondisi pencahayaan di lokasi kerja.
5. Lokasi bengkel memiliki ukuran 8 meter × 14 meter.
6. Menggunakan satu unit webcam Logitech Brio 100 dengan spesifikasi 1080p/30 FPS, kamera dipasang di atas pintu workshop pada ketinggian 2,5 meter dengan sudut kemiringan 20 derajat.

1.5 Keaslian Penelitian

Orisinalitas penelitian ini diperlukan sebagai bukti tidak adanya plagiarisme terhadap penelitian terdahulu. Tesis yang berjudul "*Prototype* detektor perlengkapan kesehatan dan keselamatan kerja (K3) berbasis *deep learning* YOLO" belum pernah diteliti dalam penelitian sebelumnya. Keaslian penelitian ini dibuktikan dalam tabel 1.1 berikut:

Tabel 1.1 Perbedaan dengan penelitian-penelitian sebelumnya

No	Judul	Metode	Hasil	Permasalahan
1	<i>Implementation Of the Yolov8 Method To Detect Work Safety Helmets</i> [2]	Akuisisi citra sebanyak 670 citra, pelabelan citra, <i>preprocessing</i> , augmentasi di roboflow, pelatihan model	mAP50 mencapai 99,5%. Sedangkan hasil pengujian dengan 100 citra, didapatkan akurasi mencapai 99%.	Faktor pencahayaan saat pengambilan citra. Selain itu, perlu ditambahkan lebih banyak

		YOLOv8x dengan 100 epochs.		<i>preprocessing</i> dan citra dengan jarak objek yang lebih jauh lagi.
2	Sistem Pendeteksi Alat Pelindung Diri (APD) pada Pekerja Konstruksi Berbasis <i>Convolutional Neural Network</i> [3]	Menggunakan kamera dan algoritma deteksi objek YOLOv7 berbasis <i>Convolutional Neural Network</i>	Mencapai akurasi yang baik pada jarak ideal 3 meter dan 4,5 meter dengan nilai mAP 0,912 dan 0,947 masing-masing. Pengujian mekanisme alarm menunjukkan akurasi sebesar 1,0 pada jarak 1,5 meter dan 3 meter.	Akurasi sistem dalam mendeteksi objek kaca mata dan sarung tangan masih kurang memenuhi ketentuan dalam memberikan pengawasan dan peringatan pada penggunaan APD
3	Implementasi Metode YOLO pada Deteksi Pakaian Keselamatan yang Lengkap di Proyek Kontruksi [4]	Sistem pengambilan keputusan menggunakan metode YOLOv5. 140 data gambar yang terdiri dari data 96 train set,	Berhasil mendeteksi dengan baik dan benar citra menggunakan pakaian keselamatan dan citra tidak menggunakan pakaian keselamatan dengan akurasi 83%.	Dataset dikumpulkan sebanyak mungkin dan bermacam-macam sudut, Hitungan area pada gambar saling

		28 valid set dan 16 test set.		bertumpuk mempengaruhi nilai akurasi dan tidak terdeteksi.
4	Rancang Bangun Aplikasi Deteksi Alat Pelindung Diri (APD) untuk Pekerja Proyek dengan Menggunakan Algoritma YoloV5 [5]	Menggunakan algoritma YoloV5. Tahap pembuatan aplikasi deteksi proses problem <i>scoping</i> , data <i>acquisition</i> , data <i>exploration</i> , modeling, <i>evaluation</i> , dan <i>deployment</i> .	Hasil persentase yang diberikan sangat bagus, yaitu helm 96%, rompi 96%, masker 95%, dan Sepatu 92% dan mAP sebesar 93,4%.	Faktor pencahayaan saat pengambilan citra. Selain itu, perlu ditambahkan lebih banyak preprocessing dan citra dengan jarak objek yang lebih jauh lagi
5	Deteksi Kelalaian Alat Pelindung Diri (Apd) Pada Pekerja Kontruksi Bangunan [6]	Pengumpulan data citra lalu pra-proses data citrai yang terdiri dari pelabelan dan perubahan ukuran citra. Kemudian dilakukan konfigurasi jaringan YoloV4	Objek pada citra APD berhasil dikenali dengan akurasi 63,03% sampai 99,4% menggunakan GPU yang kecepatan tinggi dan memiliki akurasi rata-rata 80% dengan validasi mAP sebesar 99,7%	Faktor pencahayaan, GPU saat pengambilan citra. Selain itu, perlu ditambahkan lebih banyak preprocessing dan citra dengan jarak

			dan avg loss sebesar 1.305%.	objek yang lebih jauh lagi
--	--	--	------------------------------	----------------------------



BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian M.E. Laily pada tahun 2023 membahas tentang pendeteksian penggunaan Alat Pelindung Diri (APD) di sektor konstruksi menggunakan metode *Mask Region Convolutional Neural Network* (Mask R-CNN). Tujuan untuk mendeteksi otomatis objek APD seperti helm keselamatan dan rompi menggunakan metode Mask R-CNN. Dengan pendeteksian yang otomatis, diharapkan dapat meningkatkan kepatuhan pekerja terhadap penggunaan APD, sehingga risiko kecelakaan kerja dapat diminimalisir. Metode Mask R-CNN digunakan karena kemampuan segmentasi objek yang lebih baik dibandingkan metode R-CNN sebelumnya. Mask R-CNN dapat memisahkan objek yang berbeda bahkan dalam satu kelas yang sama pada gambar atau video, dan menghasilkan output berupa mask yang bermanfaat dalam segmentasi objek. Penelitian ini mengimplementasikan Mask R-CNN dengan parameter terbaik pada epoch 35, dengan nilai loss 0,1985 dan nilai val_loss 0,1933 dalam waktu 461 detik per langkah. Uji coba dilakukan dengan 250 gambar yang menghasilkan akurasi sebesar 95% [7].

Penelitian oleh M.I. Dinata pada tahun 2023 membahas implementasi metode *Mask Region Convolutional Neural Network* (Mask R-CNN) untuk mendeteksi penggunaan Alat Pelindung Diri (APD) dalam lingkungan kerja. Mask R-CNN dipilih karena kemampuannya yang superior dalam segmentasi objek dibandingkan metode lain seperti YOLO. Fokus utama penelitian ini adalah mendeteksi empat kelas objek, yaitu helm, rompi, rambut, dan tanpa rompi, yang merepresentasikan penggunaan dan tidak penggunaan APD. Metode menggunakan dataset gambar yang diperoleh dari berbagai sumber termasuk situs penyedia gambar gratis dan situs Roboflow. Dataset yang digunakan terdiri dari 10.000 gambar sepatu yang dibagi menjadi 4 kelas, dan 2.000 gambar untuk validasi.

Dataset tersebut kemudian melalui proses pelatihan (*training*) dan pengujian (*testing*) menggunakan Mask R-CNN. Parameter utama yang digunakan dalam pelatihan adalah epoch, di mana beberapa nilai epoch diuji untuk mendapatkan hasil yang optimal. Hasil dari penelitian menunjukkan bahwa Mask R-CNN mampu mendeteksi dan membedakan empat kelas objek dengan akurasi tinggi. Model terbaik didapatkan dari parameter epoch 35 dengan akurasi sebesar 95%, presisi 96%, dan sensitivitas 97%. Dibandingkan dengan YOLO, Mask R-CNN menghasilkan deteksi objek dengan segmentasi penuh dan nilai *confidence* yang lebih tinggi [8].

Penelitian ini membahas aplikasi pendeteksi wajah pada citra digital menggunakan algoritma Haar Cascade Classifier dan transformasi citra menjadi *grayscale* dengan *library* OpenCV oleh S. Yulina pada tahun 2021. Deteksi wajah pada citra digital adalah proses penting dalam berbagai disiplin ilmu, termasuk sistem pengenalan biometrik, pencarian, dan keamanan. Komputer visi menggabungkan kecerdasan buatan dan pembelajaran mesin untuk memahami citra dan video. Penelitian ini bertujuan untuk menerapkan Haar Cascade Classifier, algoritma yang diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001, untuk mendeteksi wajah pada citra digital. Algoritma ini menggunakan fungsi cascade yang dilatih dari berbagai citra positif (dengan objek wajah) dan negatif (tanpa objek wajah). Penelitian ini juga mengubah citra berwarna menjadi *grayscale* sebagai bagian dari pemrosesan citra. Hasil penelitian menunjukkan akurasi 100% pada citra input yang memiliki objek pada posisi frontal [9].

Penelitian ini membahas tentang peningkatan pesat penggunaan UAV (*Unmanned Aerial Vehicle*) di berbagai bidang telah memunculkan masalah keamanan. Sistem deteksi yang ada saat ini seringkali kesulitan dalam identifikasi akurat, sehingga menyebabkan hilangnya deteksi drone atau alarm palsu untuk burung oleh K. Shandilya pada tahun 2023. Untuk mengatasi hal ini, dibuatlah sebuah dataset terstandarisasi yang berisi gambar drone dan burung yang beroperasi di berbagai lingkungan. Dataset ini dibuat menggunakan perangkat lunak Roboflow untuk manipulasi gambar secara efisien dengan bantuan alat berbantuan AI. Dataset

ini menjadi sumber daya yang berharga bagi para peneliti dan pengembang yang sedang mengerjakan sistem deteksi dan klasifikasi UAV yang lebih baik. Segmentasi manual yang teliti pada setiap gambar memastikan akurasi setinggi mungkin untuk melatih model-model tersebut [10].

Penelitian Saputra pada tahun 2022 bertujuan Analisis Lingkungan Kerja dan Disiplin Kerja tentang pengaruh antara variabel yang akan digunakan dalam penelitian selanjutnya, dalam lingkup Manajemen Sumber Daya. Hasil dari artikel Tinjauan Pustaka ini adalah: 1) lingkungan kerja mempengaruhi Keselamatan Kerja; 2) Lingkungan Kerja berpengaruh terhadap Kesehatan Kerja; 3) Disiplin Kerja mempengaruhi Keselamatan Kerja; 4) Disiplin Kerja berpengaruh terhadap Kesehatan Kerja; dan 5) Keselamatan Kerja mempengaruhi Kesehatan Kerja [11].

2.2 Landasan Teori

2.2.1 Kesehatan dan Keselamatan Kerja (K3)

ISO 45001:2018 Sistem Manajemen Kesehatan dan Keselamatan Kerja merupakan standar internasional yang memberikan pedoman penerapan Sistem Manajemen Kesehatan dan Keselamatan Kerja (K3), dilengkapi dengan petunjuk penggunaan agar perusahaan dapat meningkatkan keselamatan dan kesehatan kerja yang dijalankan di perusahaan. lingkungan secara proaktif dalam mencegah kecelakaan kerja dan dampak negatif aktivitas terhadap kesehatan pekerja, termasuk terjadinya penyakit akibat kerja (PAK). Penerapan ISO 45001 dirancang agar mudah diterapkan di perusahaan mana pun, apa pun ukuran atau jenisnya. Isi standar ini adalah pedoman SMK3. Selain itu, ISO 45001:2018 memungkinkan organisasi untuk secara proaktif meningkatkan kinerja SMK3 dalam mencegah cedera dan penyakit. Perlu diketahui, pemerintah mewajibkan suatu organisasi atau dunia usaha untuk menerapkan SMK3. Menurut Purwanto et al., (2020) ISO 45001 adalah standar internasional yang menetapkan standar atau persyaratan kesehatan dan keselamatan kerja / (SMK3) Sistem Manajemen K3, dengan petunjuk penggunaan, memungkinkan organisasi untuk secara proaktif meningkatkan kinerja SMK3 dalam hal pencegahan kecelakaan, penyakit dan penyakit akibat kerja [12].

Kesehatan dan Keselamatan Kerja (K3) merupakan aspek perlindungan tenaga kerja yang juga berfungsi melindungi aset perusahaan dengan tujuan utama memberikan jaminan kondisi kerja yang aman dan sehat bagi setiap karyawan serta melindungi Sumber Daya Manusia (SDM). Tujuan K3 adalah untuk mengurangi angka kecelakaan kerja, khususnya di Indonesia. Hal ini dapat tercapai jika perusahaan senantiasa memperhatikan faktor keselamatan dan kesehatan kerja, karena perhatian tersebut akan meningkatkan kinerja karyawan. Perhatian terhadap kesehatan pekerja pada awalnya lebih menekankan pada masalah keselamatan kerja, yaitu melindungi pekerja dari kerugian atau cedera yang disebabkan oleh kecelakaan kerja. Seiring dengan perkembangan industri, perusahaan mulai memperhatikan kesehatan pekerja dalam arti yang lebih luas, yakni memastikan pekerja terbebas dari penyakit fisik maupun psikologis.

2.2.2 Citra

Citra merupakan representasi atau gambaran visual dari suatu objek. Berdasarkan sifatnya, citra dapat dibagi menjadi dua kategori: citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinu, seperti gambar pada layar televisi, foto sinar-X, hasil CT scan, dan sebagainya. Sedangkan citra digital adalah citra yang dapat diproses oleh computer. Citra adalah gambar dalam dua dimensi yang terdiri dari sejumlah besar piksel, yaitu elemen terkecil dari citra. Secara umum, citra terbentuk dari kotak-kotak berbentuk persegi yang disebut piksel, dengan jarak horizontal dan vertikal antar piksel yang sama. Citra atau gambar biasanya dinyatakan dalam koordinat kartesius (x,y) , di mana setiap koordinat merepresentasikan satu bagian terkecil dari objek, yaitu piksel. Nilai pada koordinat (x,y) adalah $f(x,y)$, yang menunjukkan intensitas atau warna dari piksel pada titik tersebut.

2.2.2.1 Citra Digital

Citra digital adalah gambar yang dapat diproses oleh komputer. Secara umum, citra digital berbentuk persegi panjang atau bujur sangkar dengan lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam jumlah titik atau piksel

sehingga ukuran citra selalu berupa bilangan bulat. Setiap titik memiliki koordinat yang sesuai dengan posisinya dalam citra. Selain itu, setiap titik memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh titik tersebut.

Sebuah citra dapat didefinisikan sebagai fungsi $f(x,y)$ yang berukuran M baris dan N kolom, di mana x dan y adalah koordinat spasial, dan amplitudo f pada titik koordinat (x,y) disebut intensitas atau tingkat keabuan dari citra pada titik tersebut. Jika nilai x , y , dan nilai amplitudo f semuanya terbatas dan diskret, maka citra tersebut disebut citra digital. Gambar memperlihatkan posisi koordinat citra digital [13].



Gambar 2. 1 Koordinat Citra Digital

(Bayu, 2021)

Citra digital adalah sebuah matriks di mana indeks baris dan kolom menunjukkan suatu titik pada gambar tersebut. Citra digital terdiri atas sejumlah elemen, di mana setiap elemen merupakan elemen gambar atau piksel yang menunjukkan nilai derajat keabuan pada titik tersebut. Citra digital dengan ukuran $M \times N$ (baris = M , kolom = N) dinyatakan dalam bentuk matriks $M \times N$. Bentuk matriks citra digital adalah sebagai berikut:

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{pmatrix} \quad (2.1)$$

Berdasarkan rumus di atas, suatu citra $f(x,y)$ dapat dituliskan kedalam fungsi matematis seperti berikut ini :

$$0 \leq x \leq M - 1$$

$$0 \leq y \leq N - 1$$

$$0 \leq f(x,y) \leq G - 1$$

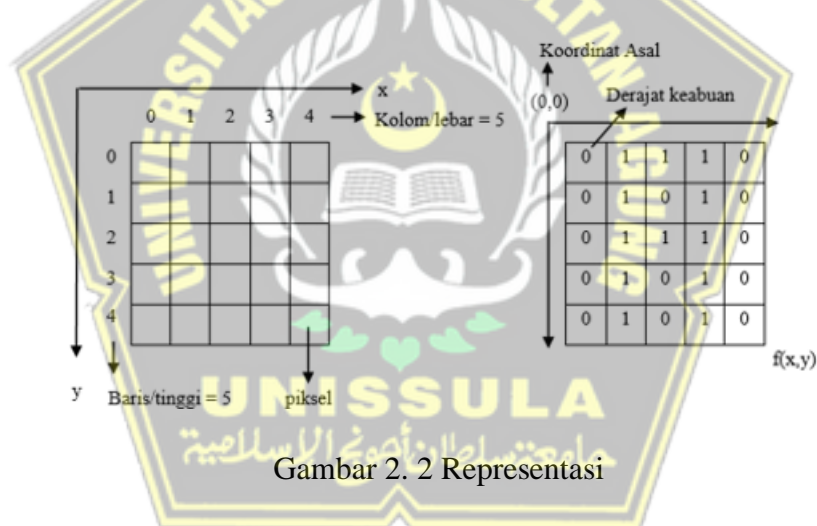
Keterangan :

N: Jumlah pixel kolom pada array citra

M: Jumlah pixel baris pada array citra

G: Nilai skala keabuan (grayscale)

Besarnya nilai G bergantung pada proses digitalisasinya. Biasanya, nilai keabuan 0 (nol) menyatakan intensitas hitam, sedangkan nilai 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan 256 warna (tingkat keabuan). Berikut ini adalah gambar yang merupakan representasi citra digital dua dimensi.



Gambar 2. 2 Representasi

(Alun, 2021)

Dari gambar di atas, terdapat nilai pada suatu potongan antara baris dan kolom (x,y) yang disebut sebagai piksel (*picture element*), elemen gambar, atau pels. Citra digital umumnya berbentuk persegi panjang dengan dimensi tertentu. Ukuran citra diukur dalam jumlah titik atau piksel, di mana setiap piksel memiliki koordinat berdasarkan letaknya dalam citra digital. Tiap piksel juga memiliki nilai yang merepresentasikan informasi yang terkandung di dalamnya [14].

2.2.3 *Deep Learning*

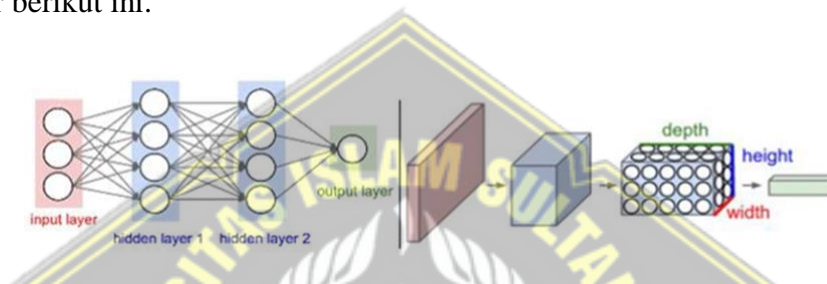
Secara definisi *deep learning* merupakan bagian dari *machine learning* yang digunakan untuk pemodelan abstraksi tingkat tinggi pada suatu data berdasarkan algoritma dengan menggunakan lapisan implementasi dan menggunakan struktur yang kompleks atau sebaliknya, terdiri dari beberapa transformasi non-linear. *Deep learning* mempunyai sebuah fitur untuk mengekstraksi pola yang didapatkan dari data yang membantu model untuk membedakan kelas sehingga fitur ini juga berperan untuk pencapaian hasil prediksi yang baik, fitur ini disebut dengan *Feature Engineering*. *Deep learning* merupakan cabang dari *Machine learning* yang terinspirasi dari korteks manusia dengan menerapkan jaringan syaraf buatan yang memiliki banyak *hidden layer*.

Machine learning menggunakan pembelajaran representasi tingkat tunggal sementara *deep learning* menggunakan kombinasi beberapa lapisan pemrosesan untuk mempelajari fitur terbaik yang diperlukan untuk mewakili data. Pembelajaran representasi tingkat yang lebih tinggi membuat *deep learning* mampu menyelesaikan tugas kompleks pada data dimensi tinggi. Sebagai contoh, dengan kekuatan komputasi yang lebih besar dan memori yang cukup besar, seseorang dapat membuat jaringan saraf dari banyak lapisan, yang disebut *deep neural network*. Keberhasilan luar biasa dalam mempelajari *deep neural network*, teknik *deep learning* saat ini berkembang lebih canggih untuk deteksi, segmentasi, klasifikasi dan recognition pada identifikasi dan verifikasi objek dalam gambar. Para peneliti sekarang bekerja untuk menerapkan keberhasilan ini dalam pengenalan pola untuk tugas-tugas yang lebih kompleks seperti diagnosa medis dan terjemahan bahasa otomatis. *Convolutional Neural Networks* adalah salah satu metode populer pada kategori *deep neural network* yang telah terbukti sangat efektif di berbagai bidang seperti pengenalan dan klasifikasi gambar.

2.2.4 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN), sebuah subbidang dari *Deep Learning*, pertama kali terinspirasi oleh penelitian Hubel dan Wiesel pada tahun

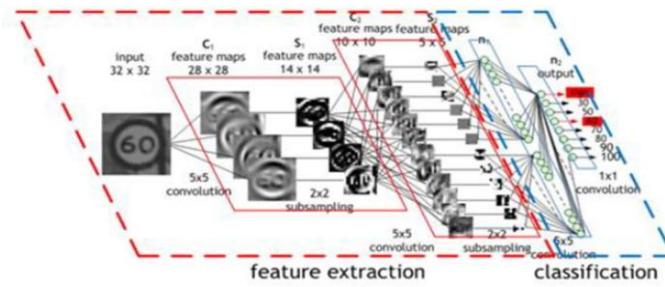
1968 mengenai korteks visual kucing. CNN merupakan pengembangan dari arsitektur jaringan saraf tiruan *multilayer perceptron* (MLP) yang dirancang khusus untuk memproses data berdimensi dua seperti citra. Meskipun MLP secara teoritis dapat digunakan untuk klasifikasi citra, CNN memiliki keunggulan dalam mempertahankan informasi spasial citra, sehingga menghasilkan kinerja yang lebih baik. Hal ini dikarenakan CNN mampu mengidentifikasi fitur-fitur lokal dalam citra secara hierarkis, sementara MLP cenderung memperlakukan setiap piksel sebagai entitas yang terpisah. Untuk memperjelas perbedaan antara jaringan saraf tiruan *multilayer perceptron* (MLP) dan *convolutional neural network* (CNN) dengan gambar berikut ini.



Gambar 2. 3 Perbedaan dimensi MLP dengan CNN

(Pramestya, 2018)

Pada gambar 2.3 *Convolutional Neural Network* (CNN) dan *Multilayer Perceptron* (MLP) memiliki prinsip kerja yang serupa, terdapat perbedaan signifikan dalam representasi neuronnya. MLP merepresentasikan setiap neuron sebagai satu dimensi, sedangkan CNN menggunakan representasi tiga dimensi. Hal ini memungkinkan CNN memproses data input berdimensi tiga, seperti gambar, secara lebih efisien. Proses transformasi dalam CNN melibatkan beberapa fungsi diferensial yang mengubah input tiga dimensi menjadi output tiga dimensi. Gambar 2.4 memberikan gambaran menyeluruh tentang struktur dan komponen-komponen utama dalam arsitektur CNN.

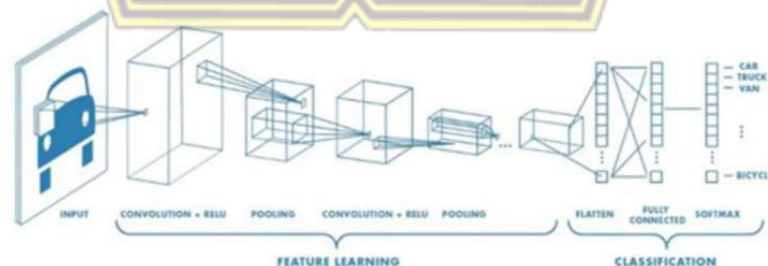


Gambar 2. 4 Contoh arsitektur CNN

(Shofia, 2018)

Gambar 2.4 memperlihatkan salah satu contoh arsitektur *Convolutional Neural Network* (CNN) yang menerima input berupa citra. Citra ini kemudian diubah menjadi vektor berdimensi 32 x 32. Setiap lapisan tersembunyi (*hidden layer*) dalam arsitektur ini memiliki sejumlah neuron yang terorganisir dalam fitur peta. Sebagai ilustrasi, lapisan C1 pada gambar tersebut terdiri dari empat fitur peta. Neuron-neuron pada lapisan C1 terhubung dengan neuron-neuron pada lapisan S1, dan demikian seterusnya hingga lapisan keluaran (*output layer*). Lapisan keluaran ini menghasilkan hasil akhir berupa klasifikasi kelas.

Convolutional Neural Network (CNN) terdiri dari beberapa lapisan yang masing-masing memiliki antarmuka pemrograman aplikasi (API) yang sederhana, seperti yang terilustrasikan pada Gambar 2.5. Secara umum, lapisan-lapisan dalam CNN dapat dikelompokkan menjadi dua bagian utama, yaitu lapisan pembelajaran fitur (*feature learning*) dan lapisan klasifikasi.



Gambar 2. 5 Lapisan pada CNN

(Shofia, 2018)

Proses pembelajaran fitur (*feature learning*) dalam suatu model terdiri dari beberapa lapisan, di antaranya lapisan konvolusi (*convolutional layers*) dan lapisan pengurangan (*pooling layers*). Lapisan-lapisan ini berperan penting dalam mengubah data input menjadi representasi numerik yang disebut fitur, yang mencerminkan karakteristik unik dari data tersebut. Lapisan konvolusi bekerja dengan cara menghitung hasil perkalian antara bobot (parameter model) dengan bagian kecil dari data input. Proses ini memungkinkan model untuk mengenali pola-pola lokal dalam data. Setiap neuron pada lapisan konvolusi akan merespons terhadap fitur-fitur tertentu pada wilayah input yang terhubung dengannya. Sementara itu, lapisan pengurangan berfungsi untuk mengurangi dimensi dari representasi fitur yang dihasilkan oleh lapisan konvolusi. Proses ini dikenal sebagai *downsampling*. Dengan mengurangi dimensi, jumlah parameter yang perlu diperbarui dalam model menjadi lebih sedikit, sehingga proses pelatihan model dapat dilakukan lebih efisien. Metode pengurangan yang umum digunakan adalah *max pooling* dan *average pooling*. *Max pooling* memilih nilai maksimum dari setiap wilayah input, sedangkan *average pooling* menghitung nilai rata-rata.

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan *feedforward* yang didesain untuk mengolah data berstruktur grid, seperti citra. Proses klasifikasi dalam CNN melibatkan beberapa tahap. Pertama, lapisan pelatakan meratakan output dari lapisan sebelumnya menjadi vektor satu dimensi. Selanjutnya, lapisan terhubung penuh melakukan perhitungan bobot untuk setiap kelas, menghasilkan skor kelas. Akhirnya, lapisan softmax menormalkan skor kelas menjadi probabilitas, di mana kelas dengan probabilitas tertinggi dianggap sebagai prediksi model. CNN mampu belajar representasi fitur hierarkis, mulai dari fitur lokal sederhana hingga fitur global yang lebih kompleks, untuk melakukan tugas klasifikasi gambar.

Arsitektur *Convolutional Neural Network* (CNN) secara umum terbagi menjadi dua komponen utama yaitu *Feature Extraction Layer* dan *Fully-Connected Layer*.

a. *Feature Extraction Layer*

Pada tahap ini, gambar awal diubah “encoding” menjadi sekumpulan angka-angka yang disebut fitur. Proses perubahan ini dinamakan ekstraksi fitur. Proses ekstraksi fitur terdiri dari dua lapisan utama, yaitu lapisan konvolusi dan lapisan penggabungan. Setiap lapisan dilengkapi dengan fungsi aktivasi. Lapisan konvolusi menerima gambar sebagai input dan mengolahnya sehingga menghasilkan keluaran berupa vektor numerik yang kemudian akan diproses oleh lapisan berikutnya.

b. *Fully-Connected Layer*

Jaringan saraf ini terdiri dari beberapa lapisan (*layer*) yang saling terhubung secara lengkap (*fully connected*). Setiap lapisan menerima input berupa vektor numerik yang merepresentasikan fitur-fitur gambar yang telah diekstraksi sebelumnya. Melalui beberapa lapisan tersembunyi (*hidden layer*), informasi tersebut ditransformasikan dan menghasilkan output berupa skor kelas, yang mengindikasikan kelas yang dimiliki oleh gambar tersebut.

2.2.5 YOLO (You Only Look Once)

You Only Look Once (YOLO) adalah suatu algoritma deteksi objek real-time yang memanfaatkan *Convolutional Neural Network (CNN)* tunggal. Dalam satu kali evaluasi, YOLO mampu memprediksi bounding box dan probabilitas kelas objek secara simultan dengan mempertimbangkan seluruh fitur gambar. Pendekatan ini dapat deteksi objek yang efisien dan akurat. Model jaringan saraf ini memanfaatkan semua informasi dalam gambar untuk memperkirakan posisi dan keberadaan objek dalam setiap kotak pembatas. Proses prediksi untuk semua kotak pembatas dan probabilitasnya dilakukan secara simultan dalam satu kali evaluasi [15].

Precision merupakan metrik evaluasi yang mengukur kemampuan sistem dalam mengklasifikasikan objek sebagai rompi atau helm dengan benar. Nilai presisi diperoleh dari perbandingan antara jumlah objek yang benar-benar diklasifikasikan sebagai rompi atau helm (*true positive*) dengan total jumlah objek yang diprediksi sebagai rompi atau helm (*predicted positive*) [16]. Perhitungan presisi secara matematis dijabarkan dalam persamaan 3.2.

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

Recall merupakan metrik evaluasi yang mengukur kemampuan suatu sistem dalam mengidentifikasi secara benar semua instance positif dalam suatu dataset. Dalam konteks deteksi objek, *Recall* menunjukkan seberapa banyak rompi atau helm (sebagai kelas positif) yang berhasil dideteksi oleh sistem dibandingkan dengan jumlah sebenarnya dari rompi atau helm yang ada dalam data. Perhitungan *Recall* dapat dilihat pada Persamaan 3.3.

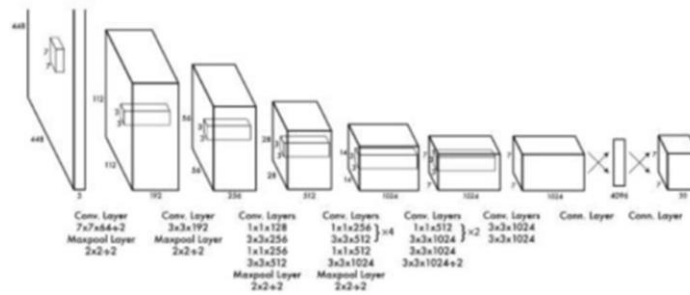
$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

Akurasi merupakan metrik evaluasi yang menunjukkan proporsi prediksi yang benar terhadap seluruh data yang telah dievaluasi. akurasi dihitung sebagai rasio dari jumlah klasifikasi yang benar (TP + TN) terhadap total jumlah klasifikasi (TP + TN + FP + FN), di mana TP (*True Positive*) adalah jumlah prediksi positif yang benar, TN (*True Negative*) adalah jumlah prediksi negatif yang benar, FP (*False Positive*) adalah jumlah prediksi negatif yang salah (diprediksi sebagai positif), dan FN (*False Negative*) adalah jumlah prediksi positif yang salah (diprediksi sebagai negatif). Persamaan 3.4 secara matematis menggambarkan perhitungan akurasi ini.

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP} \quad (2.4)$$

2.2.5.1 Desain Jaringan YOLO

Arsitektur YOLO mengandalkan 24 lapisan konvolusi untuk melakukan ekstraksi fitur. Diikuti oleh dua lapisan *fully connected* yang berperan dalam memprediksi *bounding box* dan kelas objek. Beberapa lapisan konvolusi memanfaatkan lapisan 1x1 untuk melakukan reduksi dimensi pada *feature map*, sebelum kemudian diproses lebih lanjut oleh lapisan konvolusi 3x3, seperti yang ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Arsitektur dasar YOLO

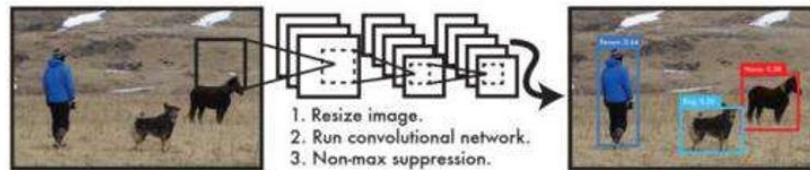
(Redmon et al, 2016)

Arsitektur YOLO mengadopsi fungsi aktivasi linier eksklusif pada output layer, sementara seluruh lapisan konvolusionalnya mengimplementasikan fungsi aktivasi leaky rectified seperti berikut.

$$\varnothing(x) = \begin{cases} x, & \text{jika } x > 0 \\ 0,1x & \text{yang lain} \end{cases} \quad (2.5)$$

Algoritma YOLO menjalankan deteksi objek melalui tiga tahap berurutan, sebagaimana yang tergambar secara visual pada Gambar 2.7.

1. *Pre-processing* guna melakukan normalisasi dimensi input gambar dengan menerapkan teknik resampling (*resize*) sehingga diperoleh representasi gambar berukuran 448 x 448 piksel. Proses ini bertujuan untuk menstandarisasi ukuran input sebelum diteruskan ke tahap ekstraksi fitur.
2. Inferensi Model untuk menerapkan model jaringan saraf tiruan (*Artificial Neural Network/ANN*) tunggal pada citra input yang telah dinormalisasi. ANN akan melakukan ekstraksi fitur dan klasifikasi objek dalam gambar berdasarkan bobot dan bias yang telah dilatih sebelumnya.
3. *Post-processing* guna melakukan *post-processing* pada output deteksi dengan menerapkan ambang batas (*thresholding*) berdasarkan nilai kepercayaan (*confidence score*). Nilai ambang batas ini akan menentukan objek mana yang dianggap terdeteksi dengan cukup akurat dan akan dipertahankan, sementara objek dengan nilai kepercayaan di bawah ambang batas akan ditolak.



Gambar 2. 7 Sistem deteksi YOLO

(Redmon et al, 2016)

Dalam metode YOLO (You Only Look Once), gambar input dibagi menjadi grid berukuran $S \times S$. Jika pusat sebuah objek berada di dalam salah satu sel grid, maka sel tersebut bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid memprediksi B kotak pembatas (*bounding boxes*) beserta nilai keyakinan (*confidence score*) untuk setiap kotak, serta probabilitas kelas kondisional C . *Confidence score* ini mencerminkan tingkat akurasi perkiraan kotak pembatas. YOLO mendefinisikan nilai *confidence* sebagai $\Pr(\text{Object}) \times \text{IoU}$. Jika tidak ada objek yang terdeteksi dalam sel, nilai *confidence* akan bernilai nol. Sebaliknya, jika ada objek yang terdeteksi, nilai *confidence* sama dengan *Intersection Over Union* (IoU) antara kotak prediksi dan *ground truth*.

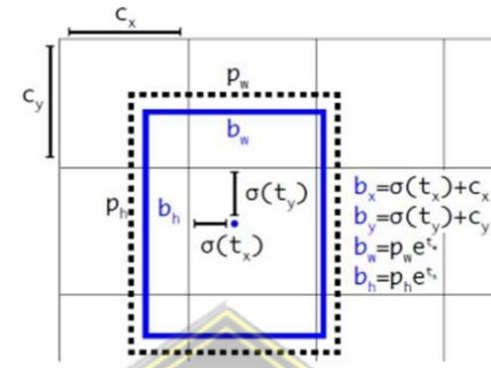
Faktor penentu dalam memperoleh prediksi akhir adalah *class confidence score*, yang didasarkan pada probabilitas kondisional kelas dan *box confidence score*. *Class confidence score* mengukur tingkat keyakinan terhadap hasil klasifikasi dan lokalisasi objek. Skor ini memberikan nilai keyakinan untuk setiap kelas spesifik terkait dengan setiap *bounding box*, yang menggambarkan probabilitas kemunculan kelas tersebut dalam kotak prediksi serta tingkat kesesuaian kotak dengan objek yang terdeteksi. Persamaan yang digunakan untuk menghitung *class confidence score* untuk setiap kotak prediksi ditunjukkan dalam persamaan berikut.

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IoU}_{\text{Pred}}^{\text{Truth}} = \text{r}(\text{Class}_i) * \text{IoU}_{\text{Pred}}^{\text{Truth}} \quad (2.6)$$

Keterangan:

$\Pr(\text{Class}_i | \text{Object})$: probabilitas kondisional kelas i .

$\Pr(\text{Object})$: Probabilitas kelas i



Gambar 2. 8 Bounding Box pada YOLO

(Redmon dan Farhadi, 2018)

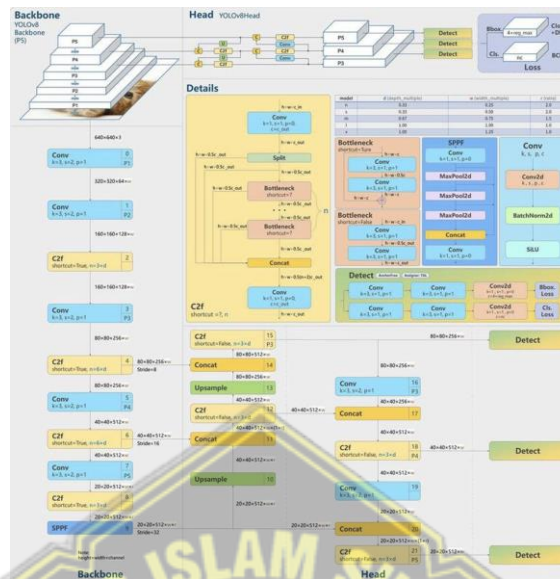


Gambar 2. 9 Model YOLO

(Redmon dan Farhadi, 2018)

Berdasarkan persamaan di atas, diperoleh nilai *confidence* untuk kelas tertentu, yang menggambarkan probabilitas kemunculan kelas tersebut dalam sebuah *bounding box* serta tingkat akurasi prediksi kotak tersebut. Sebagaimana ditunjukkan pada Gambar 2.9, model berfungsi sebagai regresi, di mana grid dibagi dan setiap grid bertanggung jawab untuk memprediksi *bounding box*, nilai *confidence* setiap kotak, serta probabilitas kelas yang bersangkutan.

Struktur internal dari algoritma YOLOv8 yang kompleks dapat dipahami lebih lanjut melalui visualisasi yang disajikan pada gambar berikut.



Gambar 2. 10 Gambar Arsitektur Jaringan YOLOV8

Gambar 2.10 memberikan representasi visual dari struktur algoritma YOLOv8 yang inovatif, yang dirancang untuk mencapai kinerja optimal dalam tugas pelatihan dan deteksi objek. Secara garis besar, arsitektur YOLOv8 terbagi menjadi dua komponen utama yang saling melengkapi: jaringan backbone, neck dan head. YOLOv8 menggunakan backbone yang canggih untuk mengekstrak fitur-fitur dari citra input. Salah satu backbone yang digunakan adalah CSPDarknet53, yang merupakan modifikasi dari arsitektur Darknet. Modifikasi ini melibatkan Cross Stage Partial networks (CSP), yang meningkatkan kemampuan belajar dan efisiensi model.

Bagian neck dalam arsitektur YOLOv8 bertanggung jawab untuk menggabungkan informasi multi-skala. YOLOv8 menggunakan *Path Aggregation Network* (PANet), yang memfasilitasi aliran informasi di berbagai skala. PANet meningkatkan kemampuan model untuk mendeteksi objek dengan ukuran yang berbeda-beda dengan lebih efektif. Dengan kata lain, YOLOv8 mampu mendeteksi objek kecil, sedang, dan besar dengan tingkat akurasi yang tinggi berkat adanya piramida fitur ini. Komponen kepala merupakan bagian akhir dari arsitektur

YOLOv8 yang bertanggung jawab untuk menghasilkan output akhir berupa prediksi deteksi. Output dari kepala ini berupa vektor yang berisi informasi mengenai probabilitas kelas, skor kepercayaan objek, dan koordinat *bounding box*. Probabilitas kelas menunjukkan suatu objek termasuk dalam kelas tertentu, sedangkan skor kepercayaan objek mengukur tingkat keyakinan model terhadap keberadaan suatu objek dalam citra. Koordinat *bounding box*, di sisi lain, menentukan lokasi dan ukuran objek yang terdeteksi.

Bagian *head* dalam arsitektur YOLOv8 bertanggung jawab untuk memprediksi kotak pembatas (*bounding box*), skor keberadaan objek (*objectness scores*), dan probabilitas kelas untuk setiap kotak pembatas yang terkait dengan sel grid. YOLOv8 menggunakan *anchor boxes* untuk memprediksi objek dengan berbagai bentuk dan ukuran dengan efisiensi tinggi. *Bounding Box Regression* adalah *head* memprediksi koordinat *bounding box* yang menentukan lokasi objek dalam gambar. Proses ini menghitung regresi dari posisi dan ukuran kotak pembatas yang mengelilingi objek. Pada YOLOv8, algoritma regresi *bounding box* ini telah ditingkatkan untuk menghasilkan prediksi yang lebih akurat, terutama dalam menangani objek kecil maupun objek yang terletak di tepi gambar. *Class Prediction* merupakan head juga memprediksi kelas objek yang terdeteksi di dalam *bounding box*. Ini dilakukan dengan menerapkan klasifikasi pada fitur yang diambil dari *backbone* dan *neck*, untuk menentukan jenis objek (misalnya, manusia, kendaraan, hewan, dan sebagainya). YOLOv8 meningkatkan akurasi prediksi kelas dengan memperbaiki mekanisme pemilihan kelas, sehingga menghasilkan identifikasi yang lebih tepat.

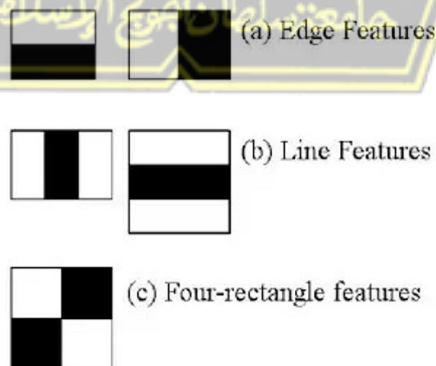
Pada YOLOv8, *confidence score* juga ditingkatkan. *Confidence score* adalah skor probabilitas yang mengukur tingkat keyakinan model bahwa objek benar-benar ada di dalam *bounding box* yang diprediksi. Peningkatan ini membantu dalam menghasilkan prediksi yang lebih andal, sehingga mengurangi jumlah prediksi yang salah (*false positives*) dan meningkatkan akurasi deteksi secara keseluruhan.

2.2.6 Haar Cascade Classifier

Metode deteksi objek menggunakan pengklasifikasian berbasis fitur Haar Cascade, yang dikembangkan oleh Paul Viola dan Michael Jones, merupakan teknik berbasis pembelajaran mesin. Dalam metode ini, fungsi cascade dilatih menggunakan sejumlah besar gambar positif dan negatif. Classifier yang digunakan dirancang untuk menganalisis gambar dengan ukuran tetap, biasanya 24 x 24 piksel. Proses deteksi dilakukan melalui teknik *sliding window*, di mana jendela berukuran 24 x 24 piksel dipindahkan secara sistematis di seluruh gambar untuk mengidentifikasi area yang menyerupai objek target. Keunggulan metode Haar Cascade terletak pada kemampuannya melakukan *scaling*, yang memungkinkan deteksi objek dengan ukuran beragam, baik kecil maupun besar, menggunakan model Haar yang sama.

2.2.6.1 Haar feature

Haar feature pada dasarnya adalah transformasi gelombang Haar dua dimensi, yang merepresentasikan sebuah gelombang persegi dengan satu interval bernilai tinggi dan satu interval bernilai rendah. Prinsip ini memanfaatkan perbedaan intensitas terang dan gelap dalam citra. Untuk meningkatkan akurasi dalam mendeteksi objek visual, beberapa kombinasi kotak digunakan. Setiap Haar feature terdiri dari gabungan area hitam dan putih yang dirancang untuk mengenali pola visual spesifik pada gambar, seperti yang ditunjukkan pada ilustrasi berikut.



Gambar 2. 10 Jenis haar feature

(Medium.com, 2024)

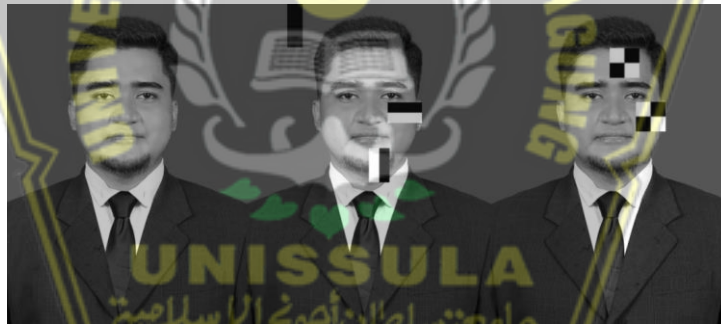
Feature:

- a. Tipe *two-rectangle feature (horizontal atau vertical)*
- b. Tipe *three-rectangle feature*
- c. Tipe *four-rectangle feature*

Penentuan fitur Haar dilakukan melalui perhitungan selisih antara rata-rata nilai piksel pada area gelap dan rata-rata nilai piksel pada area terang. Apabila selisih tersebut melebihi ambang batas yang telah ditetapkan, maka fitur tersebut dianggap terdeteksi. Nilai fitur Haar-like dihitung sebagai selisih antara jumlah nilai piksel pada area kotak berwarna hitam dan area kotak berwarna putih, sebagaimana dinyatakan dalam persamaan berikut.

$$f_x = \sum \text{black retagle} - \sum \text{white retagle} \quad (2.7)$$

Dimana untuk kotak pada Haar-like-feature dapat dihitung dengan cepat dengan fungsi “*integral image*”.



Gambar 2. 11 Jenis jenis feature dalam Haar-Like Feature

Pada setiap fitur terdapat beberapa pixel yang akan digunakan untuk melakukan penghitungan selisih antara kotak gelap dengan kotak terang pada sebuah gambar. Apabila nilai dari selisih kotak terang dan kotak gelap diatas nilai ambang maka daerah tersebut dinyatakan memiliki fitur. Ekstraksi fitur pada citra melibatkan perhitungan selisih intensitas pada kelompok piksel tertentu. Jika selisih ini melampaui nilai ambang, maka fitur dianggap terdeteksi. Fitur-fitur wajah seperti hidung, mulut, dan mata seringkali memiliki intensitas lebih rendah

dibandingkan dengan area wajah lainnya. Metode *integral image* merupakan teknik yang efisien untuk menghitung nilai fitur Haar pada citra, sehingga mempercepat proses deteksi wajah

2.2.6.2 *Integral image*

Sebuah citra digital terdiri dari tiga komponen warna utama, yaitu RGB (kombinasi dari warna merah, hijau, dan biru). Berdasarkan nilai dari masing-masing komponen RGB tersebut, dapat dihitung nilai *grayscale* (tingkat keabuan) yang diperoleh melalui rumus yang dirumuskan dalam persamaan berikut.

$$\text{Grayscale pixel} = 0.2989R + 0.5870G + 0.1140B \quad (2.8)$$

Sebagai contoh, sebuah piksel pada citra memiliki nilai $R = 100$, $G = 100$, dan $B = 100$. Dengan menerapkan perhitungan menggunakan Persamaan 2.5, diperoleh nilai derajat keabuan sebesar 99,99. Citra yang telah dikonversi menjadi skala abu-abu ditunjukkan pada gambar berikut:



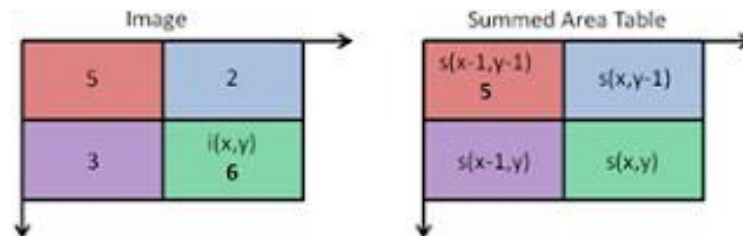
Gambar 2. 11 Perbedaan citra asil dan grayscale

(Ratna, 2020)

Metode *integral image* merupakan suatu teknik pengolahan citra yang menghitung nilai setiap piksel dalam sebuah citra sebagai akumulasi dari nilai piksel di atas dan di sebelah kirinya. Proses deteksi objek pada citra ini dilakukan dengan melakukan pemindaian menggunakan pola persegi panjang yang terdiri dari daerah terang dan gelap. Interaksi antara pola ini dengan citra akan menghasilkan fitur-fitur Haar yang digunakan untuk mengkarakterisasi objek. Perhitungan matriks *integral image* dapat direpresentasikan secara matematis melalui persamaan berikut:

$$s(x,y) = i(x,y) + s(x - 1,y) + s(x,y - 1) - s(x - 1,y - 1) \quad (2.9)$$

Perhitungan akan dilakukan dengan mempertimbangkan piksel tetangga yang memiliki koordinat (x-1, y-1), seperti yang tervisualisasi pada gambar berikut.



Gambar 2. 12 Pixel tetangga pada proses integral image

(Medium.com, 2024)

Hasil perhitungan fitur Haar akan dibandingkan dengan ambang batas (*threshold*) yang telah ditetapkan. Jika nilai fitur Haar melebihi ambang batas, maka area tersebut dianggap telah memenuhi filter Haar. Selanjutnya, area yang sama akan dievaluasi kembali menggunakan filter Haar lainnya. Apabila semua filter Haar terpenuhi, maka dapat disimpulkan bahwa terdapat objek yang sesuai pada area tersebut.

2.2.6.3 Adaboost (*Adaptive Boosting*)

Adaptive boosting merupakan suatu teknik *ensemble learning* yang bertujuan untuk meningkatkan kinerja prediksi dengan menggabungkan sejumlah *weak classifier* menjadi satu *strong classifier*. *Weak classifier* ini merupakan model prediksi yang hanya sedikit lebih baik daripada perkiraan acak. Melalui proses iteratif, algoritma *adaptive boosting* secara bertahap memberikan bobot yang lebih besar pada sampel data yang sulit diklasifikasi oleh *weak classifier* sebelumnya. Dengan demikian, *weak classifier* selanjutnya akan lebih fokus pada sampel-sampel tersebut, sehingga meningkatkan akurasi keseluruhan model.

Dalam konteks deteksi wajah, setelah tahap ekstraksi fitur Haar, fitur-fitur yang telah diekstrak akan dievaluasi kembali menggunakan algoritma *adaptive boosting*.

Tujuannya adalah untuk mengidentifikasi wilayah pada gambar yang masih mengandung fitur wajah namun belum terklasifikasi dengan baik oleh weak classifier awal. Wilayah-wilayah ini kemudian akan diproses ulang menggunakan weak classifier yang berbeda secara acak. Hasil prediksi dari *weak* classifier yang berbeda-beda ini kemudian akan dikombinasikan secara linier untuk menghasilkan keputusan akhir.

2.2.6.4 Cascade Classifier

Algoritma Cascade Classifier mengorganisasikan sejumlah besar fitur visual dalam suatu struktur klasifikasi bertingkat. Proses klasifikasi ini terdiri atas tiga tahap. Pada tahap awal, setiap sub-citra akan dievaluasi menggunakan satu fitur tunggal. Jika nilai fitur yang dihasilkan tidak memenuhi ambang batas tertentu, maka sub-citra tersebut akan langsung ditolak. Algoritma secara iteratif akan berpindah ke sub-jendela berikutnya, melakukan perhitungan ulang nilai fitur, dan membandingkannya dengan ambang batas yang telah ditetapkan. Proses ini berulang hingga diperoleh sub-jendela yang memenuhi semua kriteria filter, sehingga semakin mendekati wilayah yang mengandung wajah.

Setelah dilakukan serangkaian proses mulai dari pemilihan fitur hingga klasifikasi bertingkat, maka akan didapatkan sebuah objek deteksi berupa wajah atau bukan wajah. Pada saat proses klasifikasi berlangsung, objek yang sesuai pada gambar akan diberi kotak pada area objek tersebut. Dalam penelitian ini, Cascade Classifier yang digunakan adalah wajah Cascade Classifier depan yang telah disediakan dan dilatih oleh Google Colab.

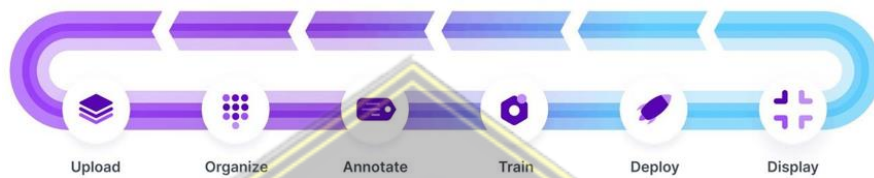
2.2.7 Roboflow

Roboflow Sebagai *platform* yang terfokus pada persiapan data visual, Roboflow menyediakan alat-alat yang efisien untuk melakukan anotasi dan pelabelan gambar secara akurat.



Gambar 2. 13 Roboflow

(Roboflow.com, 2024)



Gambar 2. 14 Alur kerja platform roboflow

(Roboflow.com, 2024)

Roboflow menawarkan solusi komprehensif bagi para pengembang kecerdasan buatan, khususnya di bidang *computer vision*. Platform ini memfasilitasi proses anotasi dan pelabelan dataset secara efisien melalui alur kerja yang intuitif. Dengan mengunggah dataset, pengguna dapat langsung mengorganisasi data dan menandai objek target dengan *bounding box*. Model-model pra-latih yang disediakan Roboflow memungkinkan pengguna untuk melatih model secara langsung dan bahkan melakukan penyebaran (*deployment*) tanpa perlu membangun infrastruktur tambahan. Roboflow menyediakan 20 *library* model yang dapat menjadi pilihan model saat akan membuat *project Computer Vision*, yaitu YOLOX, Vision Transformer, YOLOV8, YOLOV5, YOLOR, OpenAI Clip, Scaled-Yolov4, Resnet32, EfficientDet, Faster R-CNN, YOLOv3 pytorch, YOLOv3 Keras, MobileNetSSDv2, dan ResNet-32.

Platform Roboflow menyediakan fitur Health Check yang memungkinkan pengguna untuk melakukan pemantauan dan evaluasi komprehensif terhadap dataset mereka. Fitur ini dilengkapi dengan visualisasi data berupa grafik yang menampilkan distribusi kelas, jumlah data per kelas, serta perbandingan

proporsional antar kelas. Selain itu, Roboflow juga menawarkan fitur normalisasi data yang fleksibel. Pengguna dapat melakukan berbagai transformasi data seperti *scaling*, augmentasi, dan rotasi gambar untuk menyesuaikan dataset dengan kebutuhan model yang akan dibangun.

2.2.8 Google Colaboratory

Google Colaboratory, atau yang lebih dikenal sebagai Google Colab, merupakan sebuah platform berbasis web yang dirancang untuk mendukung peneliti dan pengembang dalam menjalankan kode Python langsung melalui browser tanpa perlu menginstal perangkat lunak tambahan. Platform ini dibangun di atas Jupyter Notebook dan terintegrasi dengan ekosistem Google Drive, sehingga memudahkan proses penyimpanan data serta kolaborasi antar pengguna. Popularitas Google Colab di kalangan komunitas ilmiah dan teknologi didukung oleh kemampuannya menyediakan akses gratis ke sumber daya komputasi seperti GPU dan TPU [17].

Keunggulan utama Google Colab terletak pada fleksibilitasnya dalam menangani tugas-tugas pembelajaran mesin dan analisis data skala besar. Platform ini memungkinkan pengguna untuk melakukan eksperimen dengan pustaka Python terkemuka seperti TensorFlow, PyTorch, dan Scikit-learn. Selain itu, Google Colab juga mendukung berbagai metode visualisasi data, menjadikannya alat yang sangat berguna untuk eksplorasi data serta pengembangan model deep learning. Fitur kolaborasi waktu nyata yang ditawarkan Google Colab, serupa dengan yang ada pada Google Docs, memungkinkan banyak pengguna bekerja secara bersamaan pada satu proyek. Hal ini sangat bermanfaat dalam konteks pendidikan, di mana pengajar dapat membagikan notebook kepada siswa sebagai bahan praktikum. Integrasi dengan Google Drive semakin mempermudah pengguna untuk menyimpan dan mengakses proyek mereka kapan saja dan di mana saja [18].

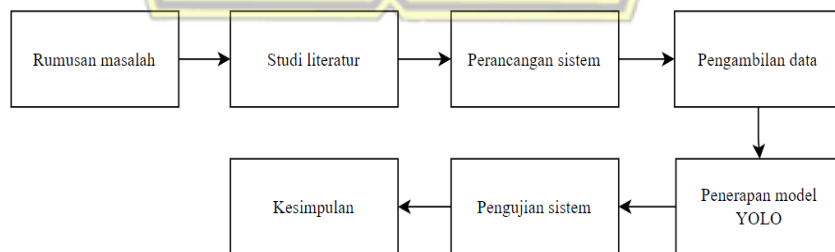
BAB III METODE PENELITIAN

3.1 Model Penelitian

Pada penelitian ini pendekatan penelitian yang digunakan adalah pendekatan penelitian kuantitatif. Pada tahap ini dilakukan beberapa observasi terlebih dahulu untuk mengetahui tahapan apa saja yang perlu dilakukan sebelum melaksanakan penelitian dan menentukan rumusan masalah yang akan diangkat dalam penelitian. Studi literatur dilakukan untuk mempelajari lebih lanjut mengenai penelitian yang akan dikerjakan sehingga mendapatkan hasil yang maksimal dan sesuai target yang diinginkan, kemudian dilanjutkan pada perancangan sistem, pengambilan data berupa gambar dan melakukan pengolahan data yang telah diperoleh. Selanjutnya melakukan penerapan pada model YOLO menggunakan data yang telah diperoleh. Setelah sistem dirasa siap dilakukan pengujian untuk mengetahui hasil kerja model yang diterapkan. Dari pengujian model dilakukan analisis sehingga menghasilkan kesimpulan dari penelitian yang dilakukan.

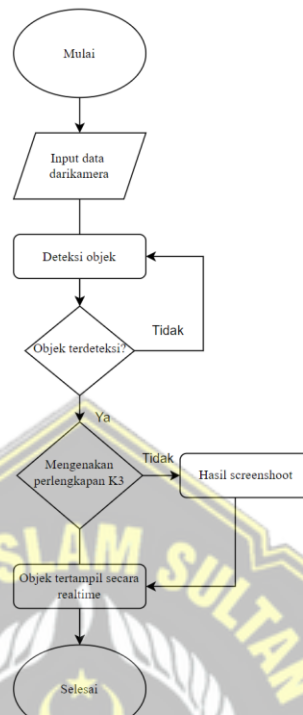
3.2 Diagram Alir

Penelitian ini dilaksanakan dengan melalui beberapa tahapan yang digambarkan dalam diagram alir seperti Gambar 3.1.



Gambar 3. 1 Diagram alir penelitian

Untuk lebih mudah dalam memahami cara kerja dari perangkat dalam penelitian kali ini maka dibuat diagram alir kerja sistem seperti pada gambar 3.2.



Gambar 3. 2 Diagram Alir kerja sistem

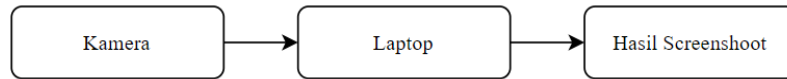
3.3 Alat Dan Bahan Penelitian

Alat yang digunakan selama kegiatan penelitian adalah sebagai berikut :

1. *Software* berupa Vscode dan bahasa python
2. Laptop Asus Vivobook Pro 14 OLED Intel Core i5
3. Webcam Logitech Brio 100 1080/30 FPS

3.4 Desain Perancangan System

Rancangan sistem yang menjadi landasan penelitian ini disajikan secara lengkap. Gambar 3.3 memberikan gambaran umum mengenai arsitektur sistem secara keseluruhan. Penjelasan mendalam mengenai setiap bagian dari sistem tersebut akan diuraikan sebagai berikut.



Gambar 3. 3 Diagram sistem keseluruhan

Berdasarkan diagram, sistem ini beroperasi dengan prinsip pengambilan data visual melalui kamera. Kamera berperan sebagai sensor input yang merekam kondisi lingkungan sekitar perangkat. Data gambar yang diperoleh kemudian diolah melalui proses pelatihan model. Model yang telah dilatih ini selanjutnya akan melakukan deteksi objek secara real-time pada perangkat. Hasil deteksi berupa visualisasi yang menunjukkan apakah pekerja telah mengenakan alat pelindung diri atau tidak. Sistem secara otomatis akan menghasilkan tangkapan layar (*screenshot*) sebagai bukti visual setiap kali deteksi menunjukkan adanya pelanggaran penggunaan alat pelindung diri.

Metode yang digunakan dalam tesis ini untuk deteksi perlengkapan kesehatan dan keselamatan kerja (K3) berbasis deep learning melibatkan beberapa tahap yang dimulai dari pengumpulan dan anotasi data menggunakan Roboflow. Dataset gambar yang mencakup helm dan rompi dalam berbagai kondisi pencahayaan, sudut, dan jarak objek, dianotasi dengan menggunakan bounding box untuk menandai objek yang relevan. Anotasi ini kemudian diimpor ke dalam platform Roboflow yang mempermudah proses persiapan dataset dan konversi ke format yang kompatibel dengan model YOLO. Proses pelatihan dilakukan di Google Colaboratory, yang menyediakan lingkungan cloud dengan GPU untuk meningkatkan efisiensi pelatihan model. Model YOLOv8 dipilih karena kemampuannya dalam deteksi objek real-time yang cepat dan akurat, serta kinerjanya yang lebih baik dalam menangani objek dengan ukuran kecil atau pada jarak jauh. Selama pelatihan, Haar Cascade digunakan untuk deteksi wajah sebagai teknik pendukung untuk mengidentifikasi area yang membutuhkan perhatian lebih, seperti memastikan bahwa pekerja mengenakan helm. Setelah model terlatih, sistem diuji dengan menggunakan webcam Brio 100 untuk menangkap gambar langsung di lingkungan kerja, kemudian hasil deteksi dievaluasi menggunakan Visual Studio Code (VS Code) pada laptop asus vivobook pro 14 OLED core i5

sebagai editor untuk mengembangkan dan menjalankan skrip deteksi. Sistem ini dapat secara otomatis mendeteksi pelanggaran K3 dengan menangkap gambar dan memberikan peringatan bila perlengkapan keselamatan tidak dikenakan oleh pekerja, meningkatkan efektivitas pemantauan keselamatan di tempat kerja.

Dalam penelitian ini terdapat beberapa variabel yang ditentukan untuk menganalisis data hasil yang dapat dirincikan sebagai berikut.

Variabel Bebas

1. Jarak dan Pencahayaan

Pada penelitian ini, variabel bebas yang diujikan adalah jarak dan pencahayaan. Kedua variabel ini dipilih karena memiliki pengaruh langsung terhadap hasil deteksi helm atau vest menggunakan sistem berbasis deep learning atau algoritma deteksi objek. Jarak merupakan variabel penting, di mana pengujian dilakukan dengan memvariasikan jarak antara objek dan sistem deteksi, mulai dari 2 hingga 12 meter. Semakin jauh jarak, detail visual objek menjadi semakin berkurang, yang dapat menurunkan akurasi sistem deteksi. Sebaliknya, pada jarak yang lebih dekat, citra yang diterima lebih jelas, sehingga akurasi deteksi cenderung lebih tinggi. Tujuan dari pengujian jarak adalah untuk mengetahui sejauh mana sistem deteksi mampu bekerja secara optimal pada berbagai jarak.

Selain jarak, pencahayaan juga menjadi faktor penting dalam deteksi objek. Kualitas citra yang diterima sistem deteksi sangat bergantung pada kondisi pencahayaan. Pada pencahayaan yang cukup, objek dapat terdeteksi dengan lebih baik, sedangkan pada pencahayaan yang rendah atau terlalu intens, sistem dapat kesulitan mengenali objek dengan benar. Penelitian ini juga bertujuan untuk mengukur sensitivitas sistem deteksi terhadap variasi pencahayaan, yang penting dalam aplikasi di lingkungan nyata dengan kondisi pencahayaan yang dinamis.

Secara keseluruhan, kedua variabel bebas ini, jarak dan pencahayaan, memainkan peran besar dalam menentukan keberhasilan sistem deteksi. Variasi jarak membantu mengevaluasi kinerja sistem pada berbagai jarak pengujian, sementara variasi pencahayaan menguji ketangguhan sistem terhadap perubahan

lingkungan cahaya. Pengujian terhadap kedua variabel ini memberikan pemahaman yang lebih mendalam tentang batasan dan kemampuan sistem deteksi yang digunakan.

Variabel Terikat

1. Akurasi

Variabel terikat dalam penelitian ini adalah akurasi deteksi helm atau vest, yang berfungsi sebagai indikator kinerja sistem deteksi berbasis deep learning atau algoritma deteksi objek. Akurasi digunakan untuk mengukur seberapa tepat sistem mengenali objek berdasarkan citra yang dihasilkan oleh kamera atau sensor dalam berbagai kondisi jarak dan pencahayaan. Semakin tinggi persentase akurasi, semakin baik sistem dalam mendeteksi objek secara benar. Akurasi ini dipengaruhi oleh variabel bebas, seperti jarak antara kamera dan objek, serta kondisi pencahayaan.

Penelitian menunjukkan bahwa seiring bertambahnya jarak, akurasi cenderung menurun karena kualitas citra dan detail visual objek semakin berkurang, yang menyulitkan sistem dalam mendeteksi objek dengan tepat. Selain itu, pencahayaan juga berperan penting, di mana pencahayaan optimal akan meningkatkan akurasi, sementara kondisi yang kurang atau berlebihan dapat menurunkan kualitas deteksi.

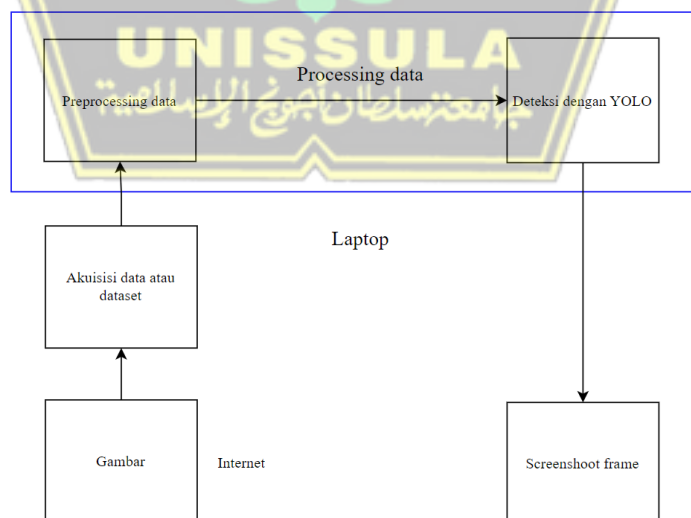
Tujuan dari pengujian akurasi ini adalah untuk mengetahui sejauh mana sistem dapat mempertahankan kinerjanya dalam berbagai kondisi, serta untuk mengevaluasi batasan dan keunggulan sistem deteksi yang digunakan. Hasil pengujian ini memberikan gambaran penting tentang efektivitas sistem dan membantu memahami kondisi di mana performa sistem menurun secara signifikan.

3.5 Sistem *processing* Data

Sistem pengolahan data yang dikembangkan dalam penelitian ini berfungsi sebagai kerangka kerja untuk mengolah kumpulan data gambar atau dataset yang telah dikumpulkan. Tujuan utama dari sistem ini adalah menghasilkan sebuah

model kecerdasan buatan yang dapat diintegrasikan dengan algoritma YOLO untuk mendeteksi perlengkapan keselamatan kerja. Proses pengolahan data dimulai dengan tahap akuisisi data, di mana gambar dikumpulkan baik melalui pengambilan gambar secara langsung menggunakan kamera maupun melalui sumber daring (internet). Penambahan data dari internet bertujuan untuk memperkaya variasi data pelatihan dan meningkatkan akurasi model yang dihasilkan.

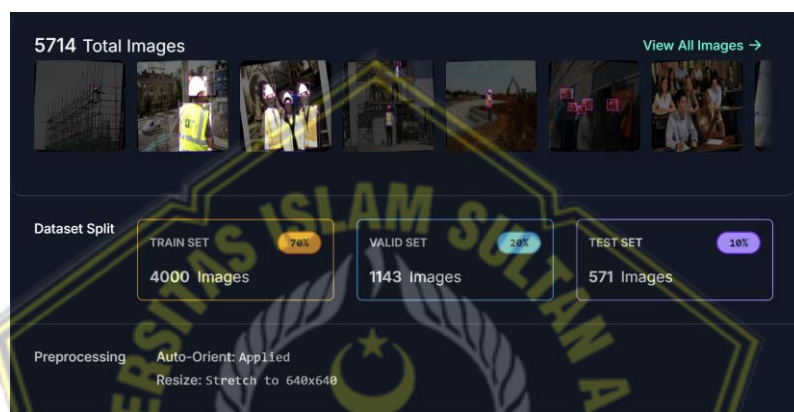
Seluruh data gambar yang telah dikumpulkan kemudian diberi label menggunakan *platform* daring Roboflow. Proses pelabelan ini sangat krusial karena label pada data akan menjadi acuan bagi model dalam mempelajari karakteristik objek yang ingin dikenali. Dengan kata lain, label akan membantu model memahami perbedaan antara satu objek dengan objek lainnya. Setelah proses pelabelan selesai, data yang telah berlabel tersebut digunakan sebagai input untuk melatih model. Selama proses pelatihan, model akan belajar secara bertahap untuk mengenali objek-objek yang telah ditentukan berdasarkan label yang diberikan. Model yang telah dilatih dengan baik kemudian akan diintegrasikan dengan algoritma YOLO. Kombinasi antara model yang telah dilatih dan algoritma YOLO ini akan menghasilkan sebuah sistem yang mampu mendeteksi perlengkapan keselamatan kerja secara *real-time*. Gambaran umum dari sistem pengolahan data yang dikembangkan dalam penelitian ini dapat dilihat pada Gambar 3.5.



Gambar 3. 4 Diagram blok sistem *processing* data

3.5.1 Akuisisi Data

Proses akuisisi data atau pembentukan dataset dilakukan melalui pengumpulan intensif data visual berupa individu, helm, dan rompi keselamatan. Data ini diperoleh melalui dua metode utama: pengambilan gambar secara langsung menggunakan perangkat kamera dan pengambilan gambar secara acak dari berbagai sumber daring menggunakan teknik *web scraping*. Pengumpulan data dilakukan secara ekstensif namun tetap memperhatikan relevansi dan korelasi antar data.

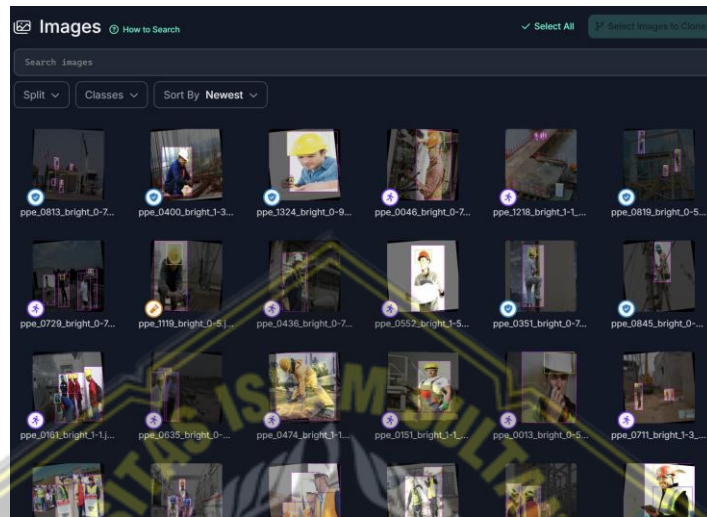


Gambar 3. 5 Akuisisi data gambar

3.5.2 Preprocessing Data

Sebelum model deteksi objek *You Only Look Once* (YOLO) dilatih, data citra yang akan digunakan terlebih dahulu menjalani serangkaian proses prapemrosesan. Tahapan prapemrosesan ini terbagi menjadi dua, yakni penyesuaian ukuran (*resizing*) dan pemotongan (*cropping*) citra, serta pemberian anotasi pada objek-objek yang terdapat di dalam citra. Tujuan dari penyesuaian ukuran dan pemotongan citra adalah untuk membuat semua data citra memiliki dimensi yang seragam dan sesuai dengan persyaratan input model YOLO, yaitu 416 x 416 piksel. Ketentuan dimensi ini telah ditetapkan dalam dokumentasi resmi YOLO. Selanjutnya pada tahap anotasi, setiap objek yang ingin dideteksi dalam citra diberi tanda berupa kotak pembatas (*bounding box*). Proses pemberian tanda ini sangat penting karena akan menjadi acuan bagi model YOLO dalam

mempelajari karakteristik visual dari berbagai objek. Untuk mempermudah dan meningkatkan efisiensi proses anotasi, peneliti memanfaatkan perangkat lunak anotasi berbasis web yang bernama Roboflow. Keunggulan dari Roboflow adalah kemampuannya untuk secara otomatis menyesuaikan format anotasi agar sesuai dengan kebutuhan spesifik model YOLO.



Gambar 3. 6 Anotasi menggunakan roboflow

3.6 Training

Pelatihan YOLOv8 di Google Colaboratory untuk mendapatkan model terbaik (best.pt) dilakukan melalui beberapa langkah penting. Dataset yang telah disiapkan di Roboflow yang telah diberi label pada objek yang ingin dideteksi akan digunakan untuk melatih model. Dataset ini disusun dan diekstraksi sesuai dengan format yang YOLOv8. Selanjutnya, buat notebook baru di Google Colab dan instal library yang diperlukan, terutama Ultralytics YOLOv8. Setelah itu, unggah dataset yang sudah dipersiapkan ke Google Drive atau langsung ke direktori Colab. Konfigurasi model dengan beberapa parameter pelatihan, seperti jumlah epoch 25, batch size 16. Setelah konfigurasi selesai, jalankan proses pelatihan di notebook, dan selama proses ini YOLOv8 akan otomatis menyimpan model terbaik (best.pt). Setelah pelatihan selesai, evaluasi kinerja model menggunakan dataset validasi untuk melihat akurasi dan metrik lainnya. Jika hasil yang diperoleh belum optimal,

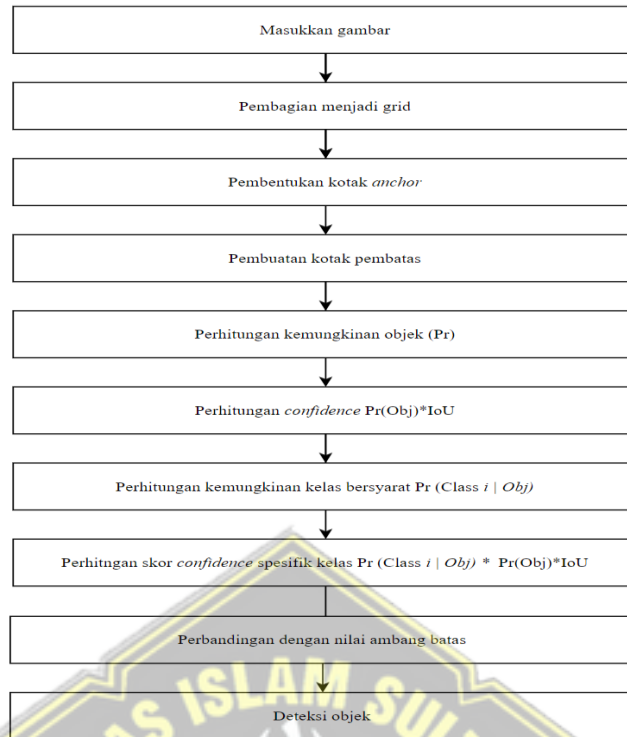
lakukan penyesuaian lebih lanjut dengan fine-tuning. Untuk meningkatkan hasil pelatihan, beberapa tips yang dapat diterapkan antara lain memanfaatkan GPU yang disediakan oleh Google Colab untuk mempercepat proses, menerapkan teknik data augmentation untuk meningkatkan variasi data dan mencegah overfitting.

3.7 Arsitektur Jaringan Yolo

Penelitian ini diawali dengan tahap persiapan model, yaitu melalui proses pelatihan menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur YOLOv8. CNN merupakan jenis jaringan saraf tiruan yang sangat populer dalam bidang penglihatan komputer, khususnya untuk tugas-tugas seperti klasifikasi dan deteksi objek. YOLOv8, sebagai salah satu varian terbaru dari keluarga algoritma YOLO, menawarkan sejumlah keunggulan seperti kecepatan pemrosesan yang tinggi dan akurasi yang baik.

3.6.1 Pendeteksian dengan Yolo

Proses identifikasi objek menggunakan algoritma YOLO dapat dijelaskan secara terperinci melalui diagram alir yang ditampilkan pada Gambar 3.7. Diagram ini memberikan ilustrasi yang terorganisasi dan sistematis tentang tahapan komputasi yang dilakukan oleh algoritma tersebut. Setiap tahapannya meliputi berbagai proses utama, dimulai dari tahap awal berupa pembacaan dan pra-pemrosesan data citra hingga tahap akhir yang mencakup identifikasi dan klasifikasi objek-objek yang terdeteksi. Diagram ini juga menunjukkan bagaimana algoritma menggunakan jaringan saraf konvolusi (CNN) untuk mengeksplorasi karakteristik citra, mendeteksi keberadaan objek, menentukan posisi objek menggunakan bounding box, serta mengelompokkan objek ke dalam kategori tertentu dengan tingkat akurasi yang tinggi. Diagram alir ini tidak hanya sekadar alat untuk memvisualisasikan proses, tetapi juga menjadi media yang menjelaskan secara mendalam bagaimana algoritma YOLO beroperasi dalam melakukan deteksi objek.



Gambar 3. 7 Diagram alir pendeteksian YOLO

Proses deteksi objek menggunakan algoritma YOLO diawali dengan pemrosesan gambar input. Gambar tersebut dibagi menjadi grid sel berukuran $S \times S$. Setiap sel grid memiliki tanggung jawab untuk mendeteksi objek tertentu dalam wilayahnya. Hasil pemrosesan awal menghasilkan kotak-kotak usulan (anchor box) yang berpotensi menampung objek. Selanjutnya algoritma menghitung probabilitas keberadaan objek dalam setiap kotak usulan dan menentukan kelas objek yang paling mungkin. Proses ini melibatkan perhitungan skor kepercayaan untuk setiap kelas objek. Skor kepercayaan ini kemudian dibandingkan dengan nilai ambang batas yang telah ditetapkan. Jika skor kepercayaan melebihi ambang batas, maka objek tersebut dianggap terdeteksi dan diklasifikasikan.

Untuk mengukur kinerja detektor objek, digunakan metrik *Intersection over Union* (IoU). IoU mengukur tingkat tumpang tindih antara kotak pembatas yang diprediksi oleh model dengan kotak pembatas yang sebenarnya (*ground truth*). Metrik ini membantu dalam mengevaluasi akurasi deteksi dan klasifikasi objek pada suatu dataset. Algoritma YOLO melakukan deteksi objek dengan cara

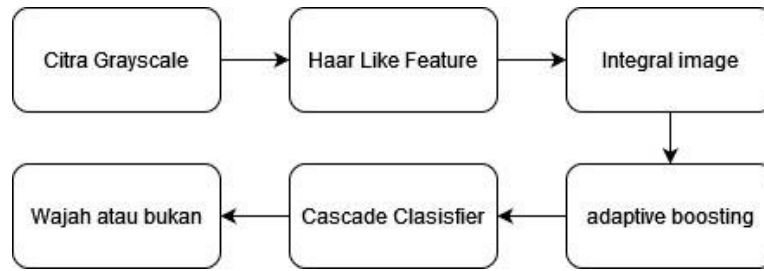
membagi gambar menjadi grid, menghasilkan kotak usulan, menghitung skor kepercayaan, dan membandingkannya dengan ambang batas. Kinerja algoritma dievaluasi menggunakan metrik IoU.

3.8 Algoritma Haarcascade Classifier

Proses deteksi objek menggunakan metode Haar Cascade Classifier melibatkan serangkaian tahap pengolahan citra yang kompleks. Sebelum menghasilkan output berupa objek terdeteksi, citra input akan melalui beberapa langkah esensial. Langkah-langkah tersebut meliputi:

1. Ekstraksi Fitur Haar-like: Fitur-fitur khas objek target (misalnya wajah) diekstrak dari citra menggunakan filter Haar-like. Fitur ini berupa kombinasi dari daerah-daerah persegi panjang dengan intensitas piksel yang berbeda-beda.
2. Perhitungan *Integral Image*: Citra diubah menjadi representasi *integral image* untuk mempercepat perhitungan nilai fitur Haar-like pada berbagai skala dan posisi.
3. Pembelajaran Adaboost: Algoritma Adaboost digunakan untuk melatih klasifikasi biner yang mampu membedakan antara objek target dan latar belakang. Klasifikasi ini akan menghasilkan serangkaian klasifikasi lemah yang kemudian digabungkan menjadi satu klasifikasi kuat.
4. Cascade Classifier: Klasifikasi kuat yang dihasilkan dari tahap Adaboost disusun secara berurutan membentuk cascade classifier. Cascade classifier ini akan memproses citra secara berurutan, dengan klasifikasi awal yang sederhana dan cepat, dan klasifikasi selanjutnya yang lebih kompleks dan akurat.

Skema lengkap dari proses deteksi objek menggunakan metode Haar Cascade Classifier dapat divisualisasikan melalui Gambar 3.8, yang menunjukkan alur pemrosesan citra mulai dari input hingga diperolehnya hasil deteksi.



Gambar 3. 8 Alur Haar Cascade Clasifier

Haar Cascade Classifier merupakan algoritma yang digunakan untuk mendeteksi objek, seperti wajah, pada citra atau video. Proses kerjanya diawali dengan mengubah citra asli menjadi citra skala abu-abu (grayscale), karena informasi intensitas dianggap cukup untuk mendeteksi fitur objek tanpa memerlukan data warna. Selanjutnya, algoritma ini menggunakan Haar-like feature, yaitu pola yang didasarkan pada perbedaan intensitas antara area terang dan gelap pada citra, seperti pola tepi atau sudut wajah. Untuk mempercepat proses perhitungan, fitur-fitur tersebut dihitung menggunakan citra integral (integral image), yang memungkinkan perhitungan rata-rata intensitas pada area tertentu dilakukan dengan efisien.

Algoritma ini kemudian memanfaatkan adaptive boosting (AdaBoost) untuk memilih fitur yang paling relevan di antara sekian banyak kemungkinan fitur, sehingga hanya fitur yang paling efektif untuk membedakan wajah dari latar belakang yang digunakan. Fitur-fitur yang terpilih tersebut kemudian diorganisasikan dalam sebuah cascade classifier, yakni struktur hierarkis yang memungkinkan pendeteksian dilakukan secara bertahap. Jika pada tahap awal suatu wilayah citra tidak memenuhi kriteria sebagai "wajah," wilayah tersebut langsung diabaikan, sehingga menghemat waktu pemrosesan. Pada tahap akhir, algoritma menentukan apakah suatu area dalam citra merupakan wajah atau bukan.

3.9 Validasi

Untuk mengevaluasi kinerja model yang telah dilatih, digunakan metrik *mean Average Precision* (mAP) yang memiliki rentang nilai antara 0 hingga 1. Nilai mAP yang mendekati 1 mengindikasikan model memiliki kemampuan deteksi yang

sangat baik. Selain mAP, confusion matrix juga digunakan sebagai alat bantu untuk menganalisis kinerja model secara lebih mendalam. Ukuran matriks konfusi pada setiap dataset bersifat dinamis, bergantung pada jumlah kelas yang digunakan dalam eksperimen. Dalam penelitian ini, matriks konfusi memiliki dimensi 3x3, yang merepresentasikan tiga kelas: rompi, helm, dan kelas lainnya (bukan keduanya). Tabel 3.1 menyajikan matriks konfusi yang diperoleh dari eksperimen sebagai berikut.

Tabel 3.1 Tabel Confusion Matrix

Jenis	Prediksi Helm	Prediksi Rompi	Bukan Keduanya
Helm	TP (Helm)	FP (Helm sebagai Rompi)	TN (Helm tidak terdeteksi)
Rompi	FP (Rompi sebagai Helm)	TP (Rompi)	TN (Rompi tidak terdeteksi)
Bukan Keduanya	FN (Bukan terdeteksi sebagai Helm)	FN (Bukan terdeteksi sebagai Rompi)	FN (Tidak terdeteksi dan benar)

Tabel 3.1 menyajikan matriks evaluasi yang menunjukkan kinerja sistem. Nilai TP (*True Positif*) mengindikasikan keberhasilan sistem dalam mengidentifikasi keberadaan rompi atau helm, sedangkan FP (*False Positif*) menunjukkan kasus di mana sistem salah mendeteksi keberadaan rompi atau helm. Nilai TN (*True Negatif*) mengindikasikan kemampuan sistem dalam mengidentifikasi dengan benar ketidakhadiran rompi atau helm dan "*False Negative* (FN) merujuk pada kondisi di mana sistem salah mengklasifikasikan suatu objek, yakni dengan menyatakan benar bukan helm atau rompi pada suatu individu dan tidak terdeteksi sebagai rompi atau helm. Keempat metrik evaluasi ini, termasuk FN, dapat digunakan sebagai acuan untuk menghitung nilai presisi, recall, dan akurasi secara keseluruhan.

3.10 Pengambilan Data dan Pengujian Data

Tahap ini merupakan tahap pengujian dimana terdapat beberapa hal yang akan diuji diantaranya pengujian secara *offline* dan pengujian secara *online* atau realtime dengan output data yang diharapkan berupa jarak, cahaya, dan akurasi deteksi objek. Tahap pengujian yang dilakukan antara lain adalah sebagai berikut:

3.10.1 Pengujian Deteksi Offline

Pengujian *offline* dilakukan dengan memberi input pada YOLO menggunakan video yang telah diambil sebelumnya. Pengujian dilakukan menggunakan spesifikasi kamera yang sama namun jarak objek dengan kamera yang berbeda beda. Hasil pengujian *offline* berupa akurasi pendeteksian objek, mAp, serta FPS (*Frame Per Second*) pada setiap jarak yang berbeda.

3.10.2 Pengujian Deteksi Online/Realtime

Pengujian secara *online* atau *Real Time* dilakukan dengan pendeteksian secara langsung menggunakan kamera yang terhubung dengan laptop, fungsi dari pengujian ini adalah untuk mengetahui kinerja sistem ketika di terapkan secara langsung. Objek yang dijadikan bahan uji akan diletakan pada jarak yan berbeda dengan data yang diharapkan berupa jarak objek, pencahayaan, akurasi, dan FPS.

3.10.3 Pengujian Keseluruhan

Pengujian keseluruhan dilakukan untuk mengetahui performa alat yang telah dibuat Ketika dilakukan pengimplementasian pada lingkungan sebenarnya. Pengujian dilakukan menggunakan metode offline dan realtime dengan input dari kamera. Pada pengujian ini ditekankan untuk mengetahui besarnya tingkat presisi alat. Hasil terbaik dari pengujian akan disajikan kedalam tabel untuk di analisis.

3.11 Lokasi Penelitian

Penelitian ini dilakukan di PT. Prima Barra Sejahtera yang terletak di Perumahan Andalusia @Regency Giri, kecamatan Kebomas, kabupaten Gresik, Jawa Timur.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 *Preparation Dataset*

Preparation Dataset adalah tahap krusial dalam pengembangan *model deep learning* yang berfokus pada persiapan data sebelum digunakan dalam proses pelatihan. Pada fase ini, data yang akan digunakan dikumpulkan, dibersihkan, dan diproses untuk memastikan kualitasnya. Proses ini dimulai dengan pengumpulan data. Kemudian, data tersebut dianotasi atau diberi label, sehingga model dapat mempelajari pola atau fitur penting dari data tersebut. Selanjutnya, data dibagi menjadi beberapa subset seperti pelatihan, validasi, dan pengujian untuk memastikan evaluasi model yang adil dan mencegah overfitting. Proses augmentasi data, seperti rotasi atau perubahan skala juga diterapkan untuk meningkatkan variasi dan keberagaman data. Dengan mempersiapkan dataset secara menyeluruh, model machine learning dapat dilatih dengan data yang berkualitas tinggi, yang berkontribusi pada kinerja dan akurasi model yang lebih baik.

4.1.1 Akuisisi Data

Dataset yang digunakan dalam penelitian ini bersumber dari kumpulan gambar yang diperoleh secara acak dari internet. Kumpulan gambar tersebut terdiri dari berbagai objek, termasuk manusia, helm, dan rompi keselamatan. Setelah dilakukan proses seleksi yang ketat, sebanyak 5.714 gambar yang memenuhi kriteria penelitian dipilih untuk tahap anotasi selanjutnya.



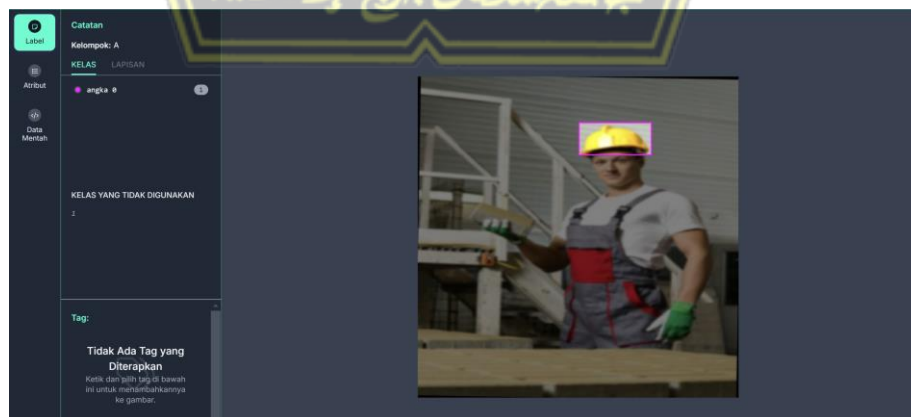
Gambar 4. 1 Gambar objek manusia menggunakan helm



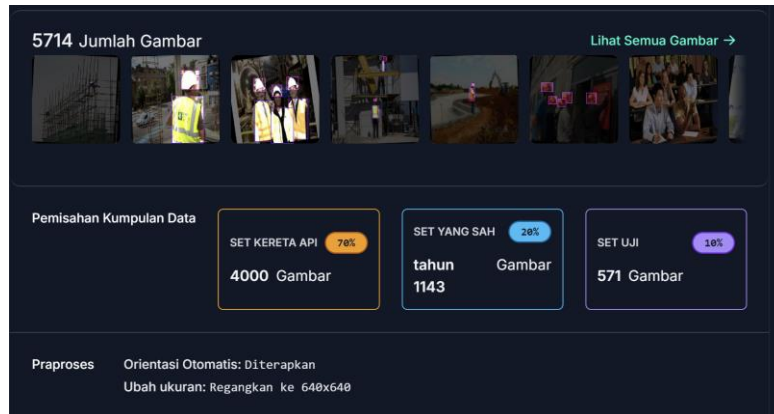
Gambar 4. 2 Gambar objek manusia menggunakan helm dan rompi keselamatan

4.1.2 Anotasi Data

Proses anotasi di Roboflow melibatkan beberapa langkah yang sistematis. Pertama, pengguna harus memilih dataset yang akan dianotasi. Dataset ini dapat berupa gambar atau video yang berisi objek yang ingin dikenali. Setelah itu, pengguna harus memberikan label pada objek tersebut dengan cara menandai objek dengan kotak batas (*bounding box*) dan menentukan kelas objek. Proses anotasi data pada dataset gambar berjumlah 5.714 ini dilakukan menggunakan platform Roboflow. Setiap objek dalam gambar diberi label dan dibatasi oleh *bounding box* (kotak pembatas). Terdapat dua kelas objek, yaitu helm (label 0) dan rompi keselamatan (label 1), dengan distribusi label masing-masing sebanyak 4.000 gambar training, 1143 gambar validasi dan testing 571 gambar. Informasi mengenai kelas objek dan koordinat *bounding box* disimpan dalam format teks (.txt). Ilustrasi proses anotasi dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Proses anotasi data dengan website roboflow



Gambar 4. 4 Persebaran dataset

4.2 Training dan Validasi

Proses pelatihan data dilakukan dengan memanfaatkan platform notebook *open-source* Google Colaboratory. Penggunaan Google Colaboratory bertujuan untuk mengoptimalkan kinerja komputasi, sehingga proses pelatihan dan validasi model dapat berlangsung lebih efisien. Platform ini menyediakan akses gratis ke unit pemrosesan grafis (GPU) yang berdaya tinggi serta integrasi penyimpanan dengan Google Drive. Proses pelatihan model dilakukan dengan cara mengkloning atau mengunduh kumpulan data yang diperlukan dari repositori GitHub YOLOv8.

```

Install YOLOv8

YOLOv8 can be installed in two ways-from the source and via pip. This is because it is the first iteration of YOLO to have an official package.

[3] # Pip install method (recommended)
!pip install ultralytics==8.2.183 -q

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()

Ultralytics YOLOv8 8.2.183 | Python 3.10.12 | torch==2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete | (3 GPU, 11.7 GB Mem, 33x711x5 GB disks)

[4] from ultralytics import YOLO

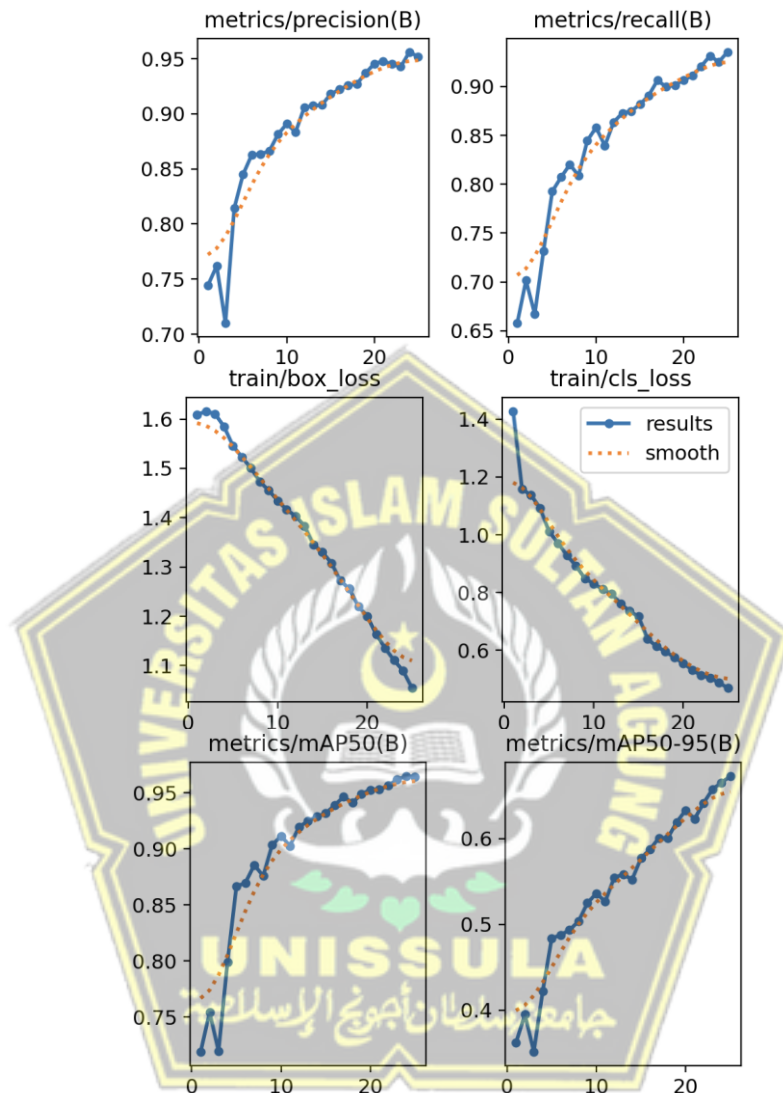
from IPython.display import display, Image
  
```

Gambar 4. 5 Training di Google Colaboratory

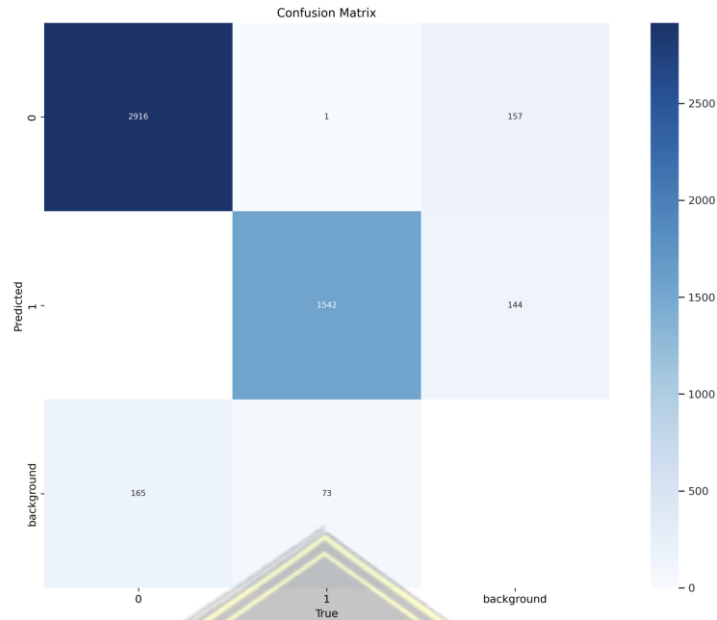
4.2.1 Model YOLOv8

Evaluasi model YOLOv8 dilakukan dengan tujuan untuk memilih model yang paling optimal dalam tugas deteksi objek. Proses evaluasi ini melibatkan penggunaan hyperparameter default yang telah ditetapkan dalam YOLOv8, yakni ukuran gambar 640 piksel, jumlah epoch 25, dan ukuran batch 16. Kombinasi hyperparameter tersebut akan diuji pada berbagai konfigurasi model YOLOv8.

Berdasarkan hasil evaluasi pelatihan model YOLOv8, diperoleh nilai rata-rata presisi (mAP@0.95) sebesar 0.67305 atau setara dengan akurasi pelatihan sebesar 67%. Visualisasi hasil dari proses pelatihan ditampilkan pada Gambar 4.6.



Gambar 4. 6 Hasil Training dengan Arsitektur Yolov8



Gambar 4. 7 Confusion matrix YOLOv8

Diketahui:

1. Perhitungan untuk helm

$$TP = 2.916$$

$$FP = 0 + 165 = 165$$

$$FN = 1 + 157 = 158$$

$$TN = 1.542 + 144 + 73 + 0 = 1759$$

Perhitungan nilai *precision* sebagai berikut,

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

$$Precision = \frac{2.916}{2.916 + 165}$$

$$Precision = 0,946 \text{ atau } 95\%$$

Perhitungan nilai *recall* sebagai berikut,

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

$$Recall = \frac{2.916}{2.916 + 158}$$

$$Recall = 0,948 \text{ atau } 95\%$$

Perhitungan nilai akurasi sebagai berikut,

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP}$$

$$Akurasi = \frac{2.916 + 1.759}{2.916 + 1.759 + 158 + 165}$$

$$Akurasi = 0.93 \text{ atau } 93\%$$

2. Perhitungan untuk rompi

$$TP = 1.542$$

$$FP = 73 + 1 = 74$$

$$FN = 0 + 144 = 144$$

$$TN = 2.916 + 157 + 165 + 0 = 3.238$$

Perhitungan nilai *precision* sebagai berikut,

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

$$Precision = \frac{1.542}{1.542 + 74}$$

$$Precision = 0,93 \text{ atau } 93\%$$

Perhitungan nilai *recall* sebagai berikut,

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

$$Recall = \frac{1.542}{1.542 + 144}$$

$$Recall = 0,914 \text{ atau } 91\%$$

Perhitungan nilai akurasi sebagai berikut,

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FN + FP}$$

$$\text{Akurasi} = \frac{1.542 + 3.238}{1.542 + 3.238 + 144 + 74}$$

Akurasi= 0.95 atau 95%

3. Perhitungan untuk bukan keduanya

TP= 0

FP= 157+144=301

FN= 165+73=238

TN= 2.916+1+0+1.542=4.459

Perhitungan nilai *precision* sebagai berikut,

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

$$\text{Precision} = \frac{0}{0 + 301}$$

Precision= 0

Perhitungan nilai *recall* sebagai berikut,

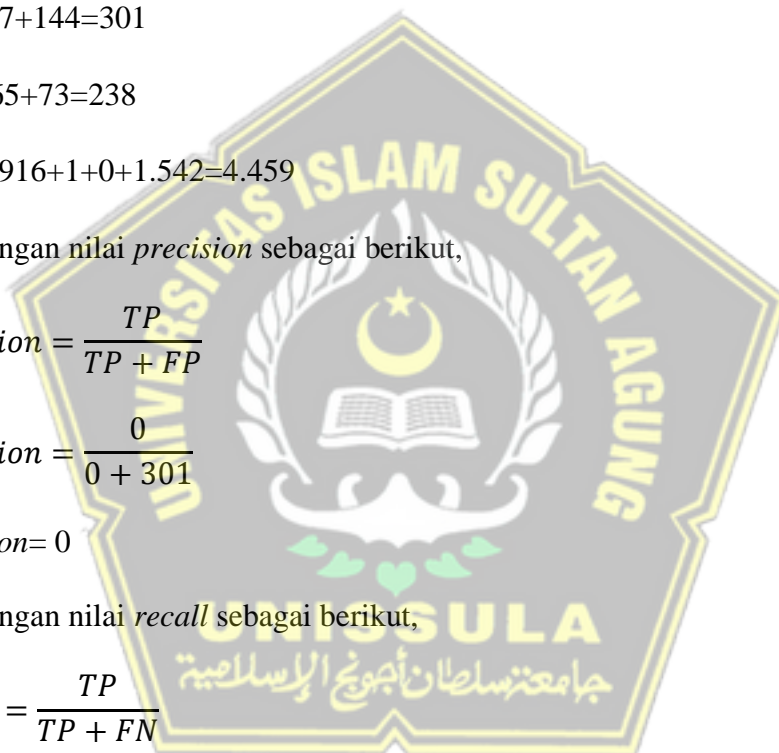
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Recall} = \frac{0}{0 + 238}$$

Recall= 0

Perhitungan nilai akurasi sebagai berikut,

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FN + FP}$$



$$\text{Akurasi} = \frac{0 + 4459}{0 + 4459 + 301 + 238}$$

Akurasi= 0.89 atau 89%

Pada gambar 4.7 *confusion matrix* yang telah dinormalisasi. Matriks ini menunjukkan kinerja model klasifikasi dengan membandingkan kelas yang sebenarnya (*True*) di sumbu horizontal dan kelas yang diprediksi (*Predicted*) di sumbu vertikal. Terdapat tiga kategori: kelas "0", kelas "1", dan "*background*" (latar belakang). Angka di dalam matriks menunjukkan persentase prediksi untuk setiap kombinasi kelas. Model memiliki akurasi sebesar 95% untuk mendeteksi kelas "0" dan "1" dengan benar, namun terdapat kesalahan sebesar 5% ketika model salah memprediksi kelas "1" sebagai "0". Demikian juga, terdapat 5% kasus di mana model salah memprediksi latar belakang sebagai objek (kelas "0" atau "1"). Matriks ini membantu dalam menganalisis apakah model kesulitan dalam membedakan antara kelas tertentu dan di mana letak kesalahan prediksi yang paling sering terjadi.

4.2.2 Nilai Batch

Batch merujuk pada jumlah sampel data yang diproses secara simultan dalam setiap iterasi pelatihan model. Parameter ini krusial dalam menentukan kualitas dan kinerja model yang dihasilkan. Dalam penelitian ini, dengan total data sebesar 5.714 sampel, penggunaan *batch size* sebesar 16 mengindikasikan bahwa algoritma akan memproses 24 sampel data pertama dalam setiap iterasi pelatihan (data ke 1, ke 2, ke 3,, ke 16) kemudian di training hingga selesai dan kembali mengambil 16 data kedua (data ke 17, ke 18, ke 19,....., ke 32) seterusnya hingga data ke 4.000. Dalam penelitian ini, telah dilakukan eksperimentasi dengan menggunakan nilai batch 16, pada proses pelatihan model YOLOv8. Model YOLOv8 ini sebelumnya telah dipilih sebagai model terbaik berdasarkan hasil evaluasi. Konfigurasi *batch size* sebesar 16 menghasilkan kinerja model yang optimal, ditandai dengan nilai mAP tertinggi. Hal ini dapat dijelaskan karena pada *batch size* yang kecil, model dilatih dengan jumlah data yang lebih terbatas, sehingga model cenderung lebih fokus pada detail data dan menghasilkan akurasi yang lebih tinggi. Selain itu, penggunaan *batch size* yang kecil memungkinkan

pemanfaatan sumber daya komputasi secara efisien selama proses pelatihan. Meningkatkan nilai ukuran batch secara signifikan dapat mengurangi akurasi model dan meningkatkan waktu pelatihan secara signifikan. Hal ini karena seiring bertambahnya ukuran batch, permintaan sumber daya komputasi juga meningkat. Berdasarkan pertimbangan antara akurasi dan efisiensi komputasi, nilai ukuran batch proses pelatihan model ditetapkan ke 16 dalam penelitian ini.

4.2.3 Nilai Epoch

Epoch merupakan parameter kunci dalam proses pelatihan model dan menentukan seberapa sering algoritma pelatihan dijalankan untuk memproses kumpulan data pelatihan. Epoch menentukan waktu pelatihan data. Epoch yang lebih banyak juga meningkatkan akurasi peta karena terdapat lebih banyak iterasi pembaruan data. Namun, penggunaan epoch yang terlalu banyak dapat membuat model menjadi bias dan membingungkan model dalam menemukan objek yang serupa dengan yang seharusnya. Dalam penelitian ini, kami menggunakan model Yolov8 dan melakukan beberapa pengujian dengan nilai epoch 25 menggunakan nilai batch 16. Hasil mAP dan akurasi tertinggi model Yolov8 diperoleh pada epoch ke-25 dengan nilai 0,965 dengan ukuran gambar 640, jumlah epoch 25, dan ukuran batch 16 memberikan performa yang sangat baik dalam hal deteksi objek. Model ini mencapai nilai mAP (*mean Average Precision*) sebesar 0.965, *Precision* (P) sebesar 0.956, dan *Recall* (R) sebesar 0.925. Performa model YOLOv8 ini menunjukkan tingkat akurasi yang tinggi dalam mendeteksi objek, ditandai dengan mAP yang mendekati sempurna, yang berarti model mampu secara konsisten mengenali objek yang benar dengan tingkat kesalahan yang sangat kecil. Selain itu, nilai *Precision* yang tinggi menunjukkan bahwa sebagian besar deteksi yang dilakukan oleh model adalah benar, sehingga model mampu meminimalkan kesalahan positif (*false positive*). Nilai *Recall* yang juga tinggi mengindikasikan bahwa model mampu mendeteksi hampir semua objek yang ada dalam data, dengan sedikit kesalahan negatif (*false negative*). Kombinasi dari mAP, *Precision*, dan *Recall* ini menunjukkan bahwa model YOLOv8 sangat andal untuk tugas deteksi objek dengan konfigurasi parameter yang digunakan.

4.3 Pengujian Secara Offline

Pengujian *offline* YOLOv8 adalah proses evaluasi performa model deteksi objek tanpa menggunakan koneksi ke server atau sumber daya eksternal. Dalam pengujian ini, dataset uji yang telah disiapkan secara lokal digunakan untuk menilai akurasi, kecepatan inferensi, dan kemampuan model dalam mendeteksi objek. Proses dimulai dengan memuat model YOLOv8 yang telah dilatih sebelumnya (pre-trained model) atau model yang dilatih secara khusus untuk tugas tertentu, kemudian model diterapkan pada dataset uji yang berisi gambar atau video. Selama pengujian offline, beberapa metrik dievaluasi, termasuk *mean Average Precision* (mAP), *Frame Per Second* (FPS). Pengujian ini bertujuan untuk memahami kinerja YOLOv8 dalam kondisi nyata, seperti deteksi objek dalam gambar statis atau rekaman video secara real-time.

4.3.1 Pengujian Deteksi Offline Model YoloV8 Menghadap Kamera






Evaluasi kinerja sistem deteksi dilakukan secara offline dengan memberikan input berupa gambar statis yang diekstrak dari video rekaman. Pengambilan video dilakukan menggunakan webcam Logitech Brio 100 dengan spesifikasi 1080p/30 FPS. Kamera diposisikan setinggi 2,5 meter dengan sudut 20 derajat. Tujuannya dilakukan tes ini untuk mengukur kemampuan sistem dalam mengidentifikasi objek pada berbagai kondisi, khususnya variasi jarak objek terhadap kamera. Eksperimen ini melibatkan serangkaian pengujian dengan jarak objek sebesar 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 dan 12 meter. Eksperimen deteksi objek secara offline ini mengadopsi model YOLOv8 yang telah dilatih sebelumnya. Model diimplementasikan pada perangkat tertentu untuk mengukur kinerja dalam hal *frame per second* (FPS), tingkat keberhasilan deteksi, dan akurasi.

4.3.1.1 Pengujian Deteksi Offline Menghadap Kamera Dengan Variasi Jarak Untuk Helm

Pengujian ini dilakukan dengan subjek yang mengenakan helm dan menghadap langsung ke kamera, di mana jarak antara subjek dan kamera divariasikan untuk mengukur akurasi deteksi. Hasil pengujian dengan variasi jarak

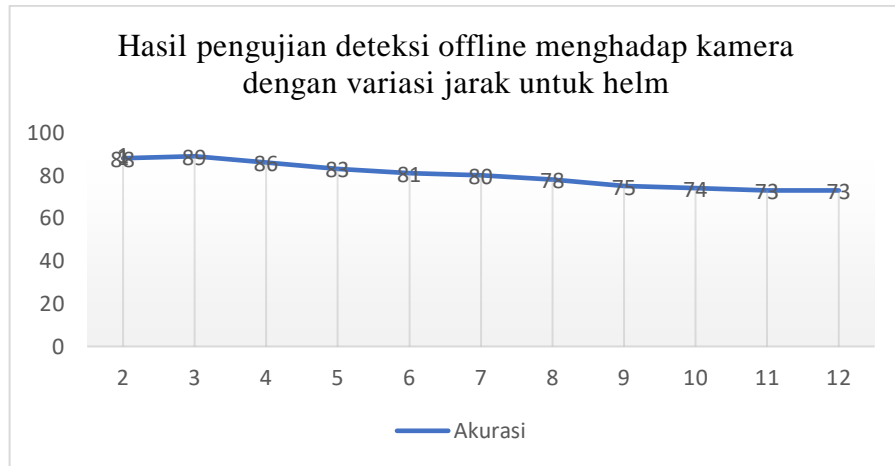
disajikan secara rinci pada Tabel 4.1.

Tabel 4. 1 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk helm

No	Objek	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Helm	2	88	3	
2	Helm	3	89	3	
3	Helm	4	86	3	
4	Helm	5	83	3	
5	Helm	6	81	3	

6	Helm	7	80	3	
7	Helm	8	78	3	
8	Helm	9	75	3	
9	Helm	10	74	3	
10	Helm	11	73	3	
11	Helm	12	73	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.8.



Gambar 4. 8 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk helm

Gambar 4.7 menggambarkan hubungan antara variasi jarak pengujian dan tingkat akurasi deteksi helm menggunakan metode tertentu. Pada sumbu horizontal, terdapat nilai jarak dari 2 hingga 12 meter, sedangkan sumbu vertikal menunjukkan tingkat akurasi dalam persentase. Pada jarak 2 meter, akurasi mencapai 88%, yang menunjukkan performa terbaik dari sistem deteksi pada jarak yang paling dekat. Hal ini bisa disebabkan oleh kualitas gambar yang lebih jelas dan informasi visual yang lebih detail pada jarak yang dekat, sehingga memudahkan sistem untuk mendeteksi helm dengan baik.


Seiring meningkatnya jarak, akurasi mulai menurun secara bertahap. Pada jarak 3 akurasi sangat baik menjadi 89%. Pada jarak 4 meter, terjadi penurunan menjadi 86%. Pada jarak 5, 6 dan 7 meter akurasi masih diatas 80%. Setelah itu pada jarak 8 meter menjadi 78%, pada jarak ini masih terlampu baik sistem dalam mendeteksi objek. Pada jarak 8 dan 10 meter terjadi penurunan menjadi 75% dan 74%. Penurunan kualitas gambar yang diambil oleh webcam atau perangkat pengambilan gambar lain karena berkurangnya cahaya atau kejernihan fokus pada jarak yang lebih jauh yang mengurangi kemampuan sistem dalam mengidentifikasi helm dengan tepat. Perubahan fluktuatif ini bisa diakibatkan oleh faktor lingkungan, kualitas gambar yang diterima pada jarak tertentu, atau bahkan variasi posisi helm pada pengujian.

Pada jarak 11 dan 12 meter, akurasi menurun sedikit menjadi 73%. Ini merupakan titik terendah dari pengujian. Penurunan pada jarak ini dapat disebabkan oleh kendala visual yang lebih besar, seperti ukuran objek yang semakin kecil dalam pandangan kamera, sehingga menyebabkan sistem kesulitan dalam melakukan deteksi secara akurat. Secara umum, grafik ini menunjukkan bahwa akurasi deteksi helm sangat dipengaruhi oleh jarak pengujian. Pada jarak yang lebih dekat, akurasi berada pada level tertinggi, sementara pada jarak yang lebih jauh, terjadi penurunan performa yang cukup signifikan. Fluktuasi di tengah grafik juga menunjukkan bahwa faktor penurunan kualitas gambar yang diambil oleh webcam atau perangkat pengambilan gambar lain karena berkurangnya cahaya atau kejernihan fokus pada jarak yang lebih jauh yang mempengaruhi hasil. Sistem deteksi ini bekerja optimal pada jarak dekat hingga menengah, namun mengalami penurunan akurasi yang tidak signifikan atau sistem berfungsi dengan baik.




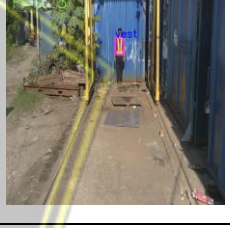
4.3.1.2 Pengujian Deteksi Offline Menghadap Kamera Dengan Variasi Jarak Untuk Vest

Pengujian ini dilakukan dengan subjek yang menggunakan vest dan menghadap langsung ke kamera, di mana jarak antara subjek dan kamera divariasikan untuk mengukur akurasi deteksi. Hasil pengujian dengan variasi jarak disajikan secara rinci pada Tabel 4.2.

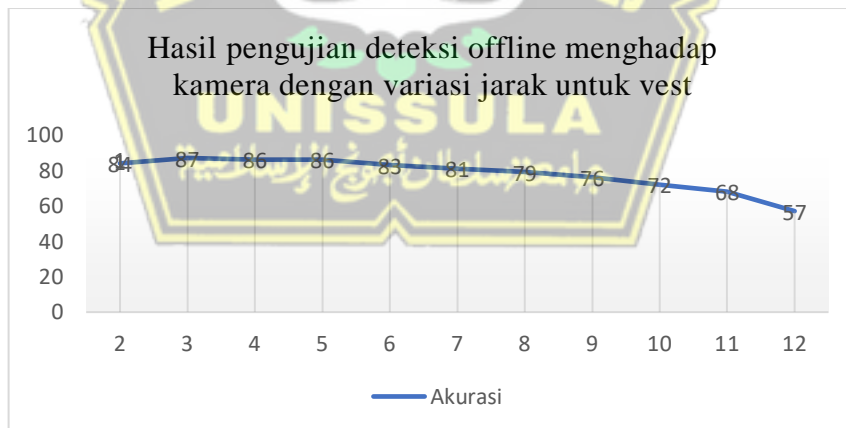
Tabel 4. 2 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk vest

No	Objek	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Vest	2	84	3	

2	Vest	3	87	3	
3	Vest	4	86	3	
4	Vest	5	86	3	
5	Vest	6	83	3	
6	Vest	7	81	3	
7	Vest	8	79	3	

8	Vest	9	76	3	
9	Vest	10	72	3	
10	Vest	11	68	3	
11	Vest	12	57	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.9.



Gambar 4. 9 Hasil pengujian deteksi offline menghadap kamera dengan variasi jarak untuk vest

Gambar 4.8 menunjukkan hubungan antara variasi jarak pengujian dengan tingkat akurasi deteksi vest (rompi) menggunakan metode tertentu. Pada sumbu

horizontal, ditampilkan variasi jarak pengujian dari 2 hingga 12 meter, sementara pada sumbu vertikal diukur tingkat akurasi dalam bentuk persentase. Pada jarak 2 meter, sistem deteksi vest menunjukkan akurasi sebesar 84%. Akurasi ini tergolong tinggi dan menandakan bahwa sistem deteksi bekerja dengan baik pada jarak dekat, di mana objek (vest) tampak jelas dan mudah dikenali oleh algoritma deteksi. Seiring dengan bertambahnya jarak, akurasi mengalami penurunan bertahap. Pada jarak 3 meter, akurasi sedikit meningkat menjadi 87%, menunjukkan performa yang tetap stabil. Namun, pada jarak 4 dan 5 meter, akurasi mulai menurun sedikit menjadi 86%, dan terus turun hingga mencapai 83% pada jarak 6 dan 81% pada jarak 7 meter.

Penurunan ini disebabkan oleh berkurangnya detail visual objek pada jarak yang lebih jauh, penurunan kualitas gambar yang diambil oleh webcam atau perangkat pengambilan gambar lain karena berkurangnya cahaya atau kejernihan fokus pada jarak yang lebih jauh yang menyebabkan sistem kesulitan dalam mengenali vest dengan tingkat ketepatan yang sama seperti pada jarak dekat. Pada jarak 8, 9 meter, akurasi terus menurun secara bertahap, dari 79% hingga 76%. Dalam jarak 10 meter terjadi penurunan akurasi menjadi 72%. Penurunan ini tergolong akurasi yang cukup akurat dicapai oleh sistem deteksi. Namun, pada jarak 11 dan 12 meter, terjadi penurunan di mana mencapai 68% dan 57%.

Secara keseluruhan, grafik ini menunjukkan bahwa sistem deteksi vest berfungsi dengan baik pada jarak dekat hingga menengah, dengan akurasi yang relatif stabil pada rentang 2 hingga 10 meter. Namun, setelah jarak 11 meter, akurasi mulai menurun signifikan. Penurunan drastis ini disebabkan oleh keterbatasan resolusi kamera, pencahayaan atau algoritma yang tidak mampu menangkap detail vest dengan jelas pada jarak yang lebih jauh. Dari hasil ini, dapat disimpulkan bahwa sistem deteksi vest efektif digunakan pada jarak kurang hingga 12 meter.

4.3.2 Pengujian Deteksi Offline Model YoloV8 Pada Posisi Dinamis

Dilakukan pengujian terhadap kemampuan model deteksi YoloV8 dalam mendeteksi objek di lingkungan kerja dengan posisi yang bervariasi. Pengujian ini


dilakukan secara offline, artinya data diolah menggunakan dataset yang telah dikumpulkan terlebih dahulu. Posisi bebas dalam konteks ini merujuk pada variasi posisi objek, seperti helm atau vest, yang tidak diatur secara spesifik, melainkan dibiarkan dalam kondisi alami sesuai dengan posisi dan aktivitas pekerja di lapangan. Tujuan dari pengujian ini adalah untuk mengetahui seberapa efektif model YoloV8 dalam mendeteksi objek dengan akurasi tinggi pada lingkungan yang dinamis, di mana posisi objek dapat berubah-ubah.







Pengujian ini penting untuk mengevaluasi kemampuan model dalam skenario dunia nyata di lingkungan kerja yang bervariasi. Hasil dari pengujian ini akan menunjukkan ketangguhan model YoloV8 dalam mengenali objek dengan berbagai posisi dan orientasi, yang mencerminkan kondisi aktual di lapangan. Selain itu, pengujian ini juga membantu mengidentifikasi tantangan yang mungkin dihadapi oleh model dalam situasi kompleks, seperti pencahayaan yang bervariasi, obstruksi sebagian, atau jarak yang berbeda-beda.




4.3.2.1 Pengujian Deteksi Offline Pada Posisi Dinamis Dengan Variasi Jarak Untuk Helm

Pengujian deteksi offline dilakukan untuk mengevaluasi akurasi sistem dalam mengenali helm pada berbagai posisi dinamis dan variasi jarak. Hasil pengujian dengan variasi jarak disajikan secara rinci pada Tabel 4.3.

Tabel 4. 3 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk helm

No	Objek	Posisi	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Helm	Menghadap kiri	2	87	3	

2	Helm	Menghadap kanan dan membungkuk	3	88	3	
3	Helm	Menghadap kiri dan membungkuk	4	87	3	
4	Helm	Menghadap kanan dan jongkok	5	80	3	
5	Helm	Menghadap kanan	6	85	3	
6	Helm	Menghadap kamera	7	80	3	
7	Helm	Menghadap kiri	8	76	3	
8	Helm	Membungkuk	9	80	3	

9	Helm	Menghadap kanan	10	73	3	
10	Helm	Menghadap kiri	11	75	3	
11	Helm	Membelakangi kamera	12	70	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.10.



Gambar 4. 10 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk helm

Berdasarkan tabel terlihat adanya fluktuasi akurasi yang dipengaruhi oleh faktor-faktor seperti pencahayaan dan posisi subjek terhadap kamera. Pada jarak 2 meter, ketika subjek menghadap ke kiri, akurasi mencapai 87%. Ini menunjukkan bahwa pada jarak yang dekat dan dengan posisi yang lebih terbuka terhadap sumber cahaya, sistem mampu mendeteksi helm dengan baik. Ketika jarak ditingkatkan

menjadi 3 meter, meskipun subjek membungkuk dan menghadap kanan, akurasi sedikit naik menjadi 88%. Peningkatan ini dapat disebabkan salah satu faktor yakni disebabkan oleh pantulan cahaya yang lebih baik atau sudut helm yang lebih mudah dikenali meskipun postur subjek lebih rumit.

Pada jarak 4 dan 5 meter, akurasi mulai menurun menjadi 87% dan 80%. Pada jarak 4 meter, subjek kembali menghadap ke kiri, namun jarak yang lebih jauh mulai mempengaruhi ketajaman citra. Pada 5 meter, posisi subjek yang jongkok membuat helm lebih sulit terdeteksi dengan jelas, yang mengakibatkan penurunan akurasi. Hal ini menunjukkan bahwa selain jarak, orientasi tubuh subjek sangat mempengaruhi performa deteksi.

Pada jarak 6 meter dengan subjek menghadap ke kanan, akurasi naik kembali menjadi 85%. Hal ini disebabkan salah satu faktor yakni oleh sudut pandang yang lebih jelas terhadap helm meskipun jaraknya lebih jauh. Namun, pada jarak 7 meter, akurasi turun lagi menjadi 80%, meskipun subjek menghadap langsung ke kamera. Penurunan ini disebabkan salah satu faktor oleh adanya bayangan atau helm yang tidak terlihat sempurna.






Pada jarak antara 8 akurasi kembali turun disebabkan posisi menghadap kiri. Pada jarak 9 meter, akurasi mengalami kenaikan karena posisi membungkuk dan posisi sejajar dengan kamera sehingga kamera dapat mendeteksi dengan akurat. Ketika jarak 10 hingga 12 meter, akurasi terus menurun dari 75% hingga 70%. Posisi subjek yang lebih tertutup, seperti menghadap kanan dan sedikit cahaya disebabkan banyak peralatan di wokshop. Pencahayaan dan orientasi subjek terhadap kamera sangat berpengaruh pada akurasi deteksi helm. Posisi yang lebih terbuka dan terpapar cahaya cenderung menghasilkan akurasi yang lebih tinggi, sementara posisi yang lebih tertutup atau tidak menghadap kamera mengurangi performa deteksi secara signifikan.

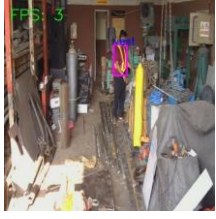


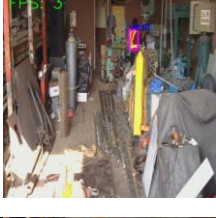
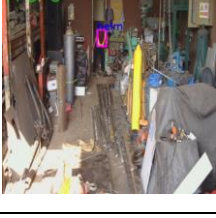
4.3.2.2 Pengujian Deteksi Offline Pada Posisi Dinamis Dengan Variasi Jarak Untuk Vest

Pengujian deteksi offline dilakukan untuk mengevaluasi akurasi sistem dalam mengenali helm pada berbagai posisi dinamis dan variasi jarak. Proses pengujian ini melibatkan pergerakan objek secara dinamis untuk mensimulasikan kondisi

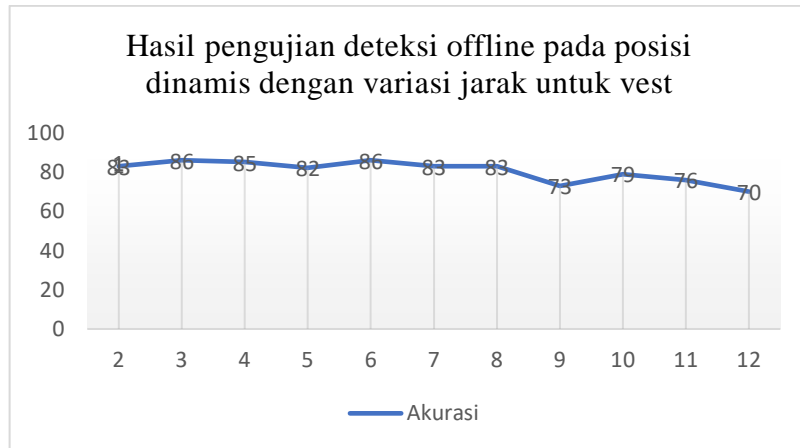
nyata di lapangan. Hasil pengujian dengan variasi jarak disajikan secara rinci pada Tabel 4.4.

Tabel 4. 4 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk vest

No	Objek	Posisi	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Vest	Menghadap kiri	2	83	3	
2	Vest	Membelakangi kamera	3	86	3	
3	Vest	Menghadap kamera	4	85	3	
4	Vest	Membelakangi kamera dan jongkok	5	82	3	
5	Vest	Menghadap kamera	6	86	3	

6	Vest	Membelakangi kamera	7	83	3	
7	Vest	Menghadap kanan	8	83	3	
8	Vest	Menghadap kiri dan jongkok	9	73	3	
9	Vest	Membelakangi kamera	10	79	3	
10	Vest	Menghadap	11	76	3	
11	Vest	Membelakangi kamera	12	70	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.11.



Gambar 4. 11 Hasil pengujian deteksi offline pada posisi dinamis dengan variasi jarak untuk vest

Analisis menyeluruh dari tabel deteksi vest menunjukkan bahwa fluktuasi akurasi dipengaruhi oleh perbedaan cahaya dan variasi posisi subjek. Pada jarak pendek (antara 2 hingga 4 meter), akurasi deteksi relatif tinggi, mencapai sekitar 83% hingga 85%. Hal ini disebabkan oleh pencahayaan yang lebih stabil pada jarak tersebut, sehingga sistem dapat menangkap detail visual dengan lebih jelas. Sebagai contoh, ketika subjek menghadap kiri pada jarak 2 meter, akurasi deteksi mencapai 83%, sedangkan pada jarak 3 meter, saat subjek membelakangi kamera, akurasi naik sedikit menjadi 86% dan ketika menghadap kamera akurasi naik menjadi 85%.

Pada saat jarak 5 meter, akurasi mulai mengalami penurunan terlihat dari angka akurasi 82%. Pada jarak 6 meter yang posisi menghadap kamera, akurasi mengalami kenaikan menjadi 86%. Pada jarak 7 dan 8 meter, akurasi menurun tetapi stabil di angka 83%. Penurunan ini disebabkan oleh variasi posisi tubuh subjek, seperti badan tidak menghadap kamera atau membelakangi kamera, yang mengakibatkan sistem lebih sulit mengenali detail visual dengan konsisten. Posisi membelakangi kamera secara terus-menerus pada jarak tersebut menurunkan jumlah detail yang dapat diidentifikasi.

Pada jarak yang lebih jauh 9 meter, posisi tubuh yang lebih kompleks, seperti jongkok atau tidak menghadap kamera, berdampak signifikan pada performa deteksi menjadi 75%. Meskipun akurasi kembali meningkat menjadi 79%

pada jarak 10 meter, kualitas pencahayaan yang menurun dan posisi subjek yang lebih menantang menyebabkan penurunan akurasi drastis hingga mencapai 76% pada jarak 11 meter. Akurasi terus berkurang dan mencapai titik terendah pada jarak 12 meter, dengan akurasi 70%, terutama ketika subjek membelakangi atau berada dalam posisi yang menyulitkan sistem mengenali pola visual.

Secara keseluruhan dari percobaan fluktuasi akurasi ini dipengaruhi oleh kombinasi jarak, perubahan posisi tubuh, dan variasi intensitas cahaya, yang secara keseluruhan memperumit proses deteksi pada jarak yang lebih jauh. Deteksi optimal terjadi pada jarak 4 dan 6 meter, dengan posisi tubuh yang lebih jelas atau menghadap kamera sementara akurasi menurun tajam saat jarak semakin jauh dan posisi tubuh semakin kompleks.

4.3.3 Analisis Pengujian Secara Offline Model YOLOv8 pada Objek dalam Posisi Menghadap Kamera dan Posisi Dinamis

Analisis hasil pengujian yang ditampilkan pada Gambar 4.7 mengenai deteksi helm dan Gambar 4.8 mengenai deteksi vest (rompi) memperlihatkan pengaruh jarak dan posisi subjek terhadap akurasi serta performa sistem deteksi berbasis metode tertentu. Pada jarak dekat, sekitar 2 hingga 3 meter, sistem deteksi helm menunjukkan performa terbaik dengan akurasi maksimal sebesar 89%. Akurasi ini dipengaruhi oleh ketajaman citra yang dihasilkan pada jarak dekat, yang memungkinkan algoritma mengenali helm secara konsisten. Faktor pencahayaan yang baik dan resolusi citra yang tinggi juga mendukung kemampuan deteksi, terutama pada jarak ini. Deteksi vest juga menghasilkan performa baik pada jarak 2 hingga 3 meter, dengan akurasi antara 83% hingga 87%, yang menunjukkan bahwa jarak dekat mengurangi gangguan visual, memungkinkan sistem berfungsi secara optimal.

Ketika bertambahnya jarak, akurasi deteksi secara bertahap menurun. Pada jarak 4 hingga 7 meter, akurasi deteksi helm masih di atas 80%, tetapi mulai mengalami penurunan akibat berkurangnya detail visual objek, kurangnya pencahayaan, dan penurunan kualitas citra yang diterima perangkat. Posisi subjek

yang semakin kompleks, seperti membungkuk atau membelakangi kamera, juga menurunkan performa deteksi karena detail visual menjadi lebih sulit dikenali. Deteksi vest menunjukkan pola serupa, dengan akurasi menurun antara 81% hingga 76% pada jarak tersebut, menunjukkan bahwa penambahan jarak berpengaruh signifikan terhadap kemampuan sistem dalam mengenali objek dengan baik.

Penurunan akurasi yang lebih tajam terjadi pada jarak 8 hingga 12 meter. Akurasi deteksi helm turun dari 78% hingga mencapai 70% pada jarak 12 meter. Faktor seperti penurunan ketajaman citra, kualitas pencahayaan yang menurun, dan objek yang semakin kecil dalam pandangan kamera menjadi penyebab utama penurunan performa deteksi pada jarak ini. Deteksi vest juga mengalami penurunan akurasi yang signifikan setelah jarak 10 meter, dengan akurasi turun dari 72% pada jarak 10 meter menjadi 57% pada jarak 12 meter, menunjukkan bahwa sistem kurang efisien dalam menangani objek pada jarak jauh ketika kualitas gambar tidak lagi memadai.

Variasi posisi subjek terhadap kamera juga memengaruhi performa sistem deteksi. Posisi yang terbuka, seperti menghadap kamera, menghasilkan akurasi lebih tinggi, sedangkan posisi tertutup, seperti membelakangi kamera atau membungkuk, menurunkan akurasi secara signifikan. Ini terlihat dari hasil pengujian, di mana pada jarak tertentu, ketika subjek berada pada posisi yang tidak ideal, akurasi turun tajam meskipun jaraknya relatif dekat. Pencahayaan yang tidak merata juga berdampak pada performa deteksi, di mana penurunan intensitas cahaya pada jarak jauh memaksa sistem bekerja lebih keras, memperlambat deteksi dan mengurangi akurasi. Secara keseluruhan dalam percobaan ini menunjukkan bahwa sistem deteksi helm dan vest bekerja optimal pada jarak dekat hingga menengah (2 hingga 8 meter) dengan akurasi tinggi dan performa stabil. Namun, pada jarak lebih jauh (9 hingga 12 meter), baik akurasi deteksi mengalami penurunan akibat kombinasi faktor jarak, pencahayaan, dan variasi posisi subjek. Peningkatan kualitas pencahayaan atau algoritma yang lebih adaptif terhadap variasi posisi dapat meningkatkan kinerja sistem secara keseluruhan.

4.4 Pengujian Secara Online atau Realtime

Pengujian realtime atau online YOLOv8 melibatkan penerapan model untuk mendeteksi objek secara langsung pada video atau stream kamera dalam waktu nyata. YOLOv8 dirancang khusus untuk aplikasi real-time dengan latensi rendah dan throughput tinggi, yang memungkinkannya memproses gambar atau frame video dalam hitungan milidetik. Pengambilan video dilakukan menggunakan webcam Logitech Brio 100 dengan spesifikasi 1080p/30 FPS. Kamera diposisikan setinggi 2,5 meter dengan sudut 20 derajat. Pengujian ini biasanya dilakukan dengan menggunakan sumber data langsung melalui webcam. Dalam pengujian ini, YOLOv8 menangkap setiap frame, mengekstraksi fitur, mendeteksi objek, dan menampilkan bounding box beserta label objek dengan cepat.

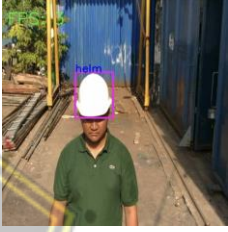



4.4.1 Pengujian Deteksi Secara Realtime Model YoloV8 Menghadap Kamera




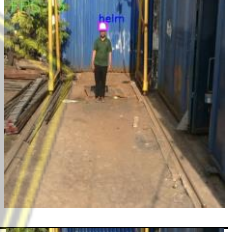



Pengujian deteksi secara realtime menggunakan model YOLOv8 dilakukan untuk mengevaluasi kemampuan sistem dalam mengenali objek ketika objek tersebut berhadapan langsung dengan kamera. Model YOLOv8, yang dikenal dengan kecepatan dan efisiensinya, diterapkan pada lingkungan pengujian dengan berbagai skenario agar dapat menangkap berbagai posisi dan ekspresi objek. Pada pengujian ini, objek diletakkan dalam jarak yang bervariasi dari kamera, dan hasil deteksi dipantau secara langsung untuk mengevaluasi performa model. Sistem diharapkan mampu memberikan hasil yang akurat meskipun terdapat perubahan posisi atau ekspresi dari objek di depan kamera. Pengambilan video dilakukan menggunakan webcam Logitech Brio 100 dengan spesifikasi 1080p/30 FPS. Kamera diposisikan setinggi 2,5 meter dengan sudut 20 derajat. Pengujian ini dilakukan dengan menggunakan input dari kamera dan menampilkan hasil deteksi secara langsung pada monitor perangkat. Variabel utama yang dimanipulasi dalam pengujian ini adalah jarak antara objek dengan kamera.

4.4.1.1 Pengujian Deteksi Secara Real-Time Saat Menghadap Kamera Dengan Variasi Jarak Untuk Helm

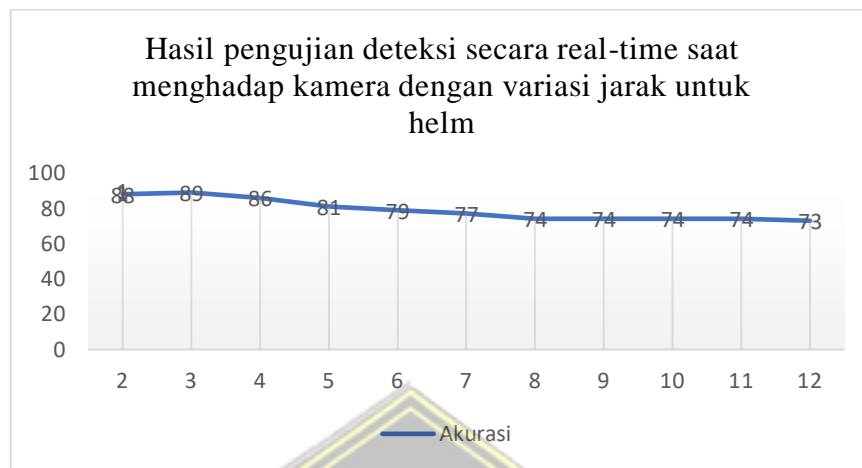
Pengujian dilakukan dengan mendeteksi penggunaan helm secara real-time pada subjek yang menghadap langsung ke kamera dengan berbagai variasi jarak. Tujuan dari pengujian ini adalah untuk mengevaluasi kinerja sistem deteksi dalam mengidentifikasi helm pada jarak yang berbeda-beda. Hasil pengujian dengan berbagai variasi jarak tersebut disajikan pada Tabel 4.5.

Tabel 4. 5 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk helm

No	Objek	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Helm	2	88	3	
2	Helm	3	89	3	
3	Helm	4	86	3	
4	Helm	5	81	3	

5	Helm	6	79	3	
6	Helm	7	77	3	
7	Helm	8	74	3	
8	Helm	9	74	3	
9	Helm	10	74	3	
10	Helm	11	74	3	
11	Helm	12	73	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.12



Gambar 4. 12 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk Helm

Gambar 4.11 merupakan hubungan antara jarak pengujian dan tingkat akurasi deteksi helm dalam skenario realtime. Pada sumbu horizontal, ditampilkan variasi jarak dari 2 hingga 12 meter, sementara pada sumbu vertikal diukur tingkat akurasi dalam bentuk persentase. Pada jarak 2 meter, akurasi awal sebesar 88% menunjukkan performa deteksi yang cukup baik, meskipun belum mencapai hasil optimal. Pada jarak 3 meter, akurasi meningkat menjadi 89%, yang merupakan tingkat akurasi tertinggi dalam pengujian ini. Jarak 3 meter dianggap ideal karena kualitas citra yang lebih optimal sehingga mempermudah deteksi.




Setelah jarak 3 meter, akurasi mulai menurun secara bertahap. Pada jarak 4, 5, 6 dan 7 meter, akurasi turun menjadi 86%, 81%, 79 dan 77%, masing-masing. Penurunan ini masih terbilang kecil, yang menunjukkan bahwa sistem tetap mampu mendeteksi helm dengan baik pada jarak tersebut meskipun kualitas citra mulai berkurang. Pada jarak 8 hingga 11 meter, di mana akurasi stabil di angka 74% meskipun sistem mulai mengalami kesulitan dalam mendeteksi helm akibat berkurangnya pencahayaan pada jarak yang semakin jauh. Pada jarak 12 meter, akurasi turun lagi menjadi 73%, yang dapat menunjukkan bahwa cahaya cukup baik


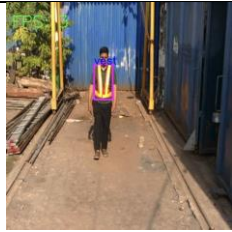
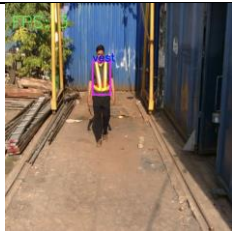


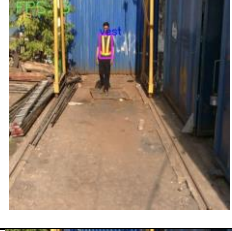
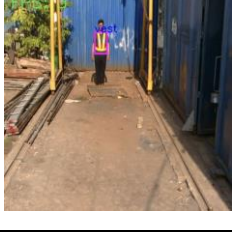
meskipun jarak cukup jauh. Secara keseluruhan, naik turunnya akurasi dalam pengujian ini dipengaruhi oleh kondisi pencahayaan yang memengaruhi kualitas citra yang diterima sistem pada jarak yang berbeda.


4.4.1.2 Pengujian Deteksi Secara Real-Time Saat Menghadap Kamera Dengan Variasi Jarak Untuk Vest

Pengujian deteksi vest secara real-time dilakukan dengan berbagai variasi jarak untuk menguji keakuratan sistem dalam mendeteksi objek saat subjek menghadap kamera. Tujuan dari pengujian ini adalah untuk mengevaluasi kinerja sistem deteksi dalam mengidentifikasi helm pada jarak yang berbeda-beda. Hasil pengujian dengan berbagai variasi jarak tersebut disajikan pada Tabel 4.6.

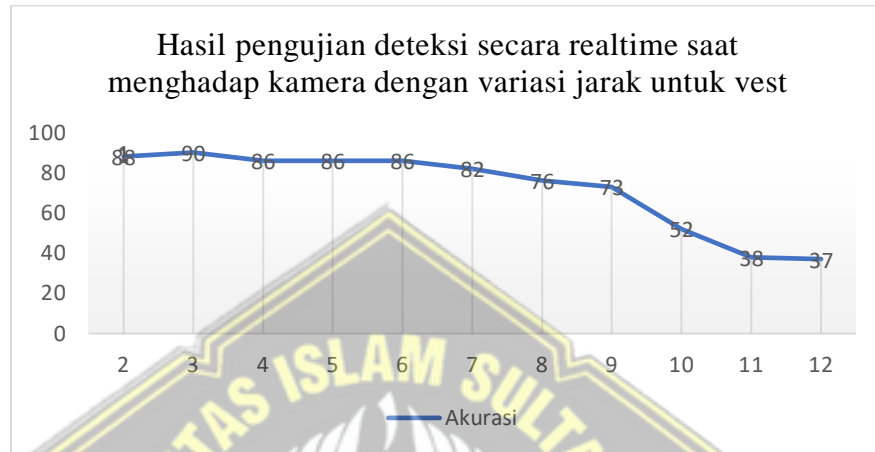
Tabel 4. 6 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk vest

No	Objek	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Vest	2	88	3	
2	Vest	3	90	3	
3	Vest	4	86	3	

4	Vest	5	86	3	
5	Vest	6	86	3	
6	Vest	7	82	3	
7	Vest	8	76	3	
8	Vest	9	73	3	
9	Vest	10	52	3	
10	Vest	11	38	3	

11	Vest	12	37	3	
----	------	----	----	---	---

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.13



Gambar 4. 13 Hasil pengujian deteksi secara real-time saat menghadap kamera dengan variasi jarak untuk vest

Gambar 4.12 yang ditampilkan menunjukkan hasil pengujian deteksi real-time dengan variasi jarak untuk sebuah objek (dalam hal ini, "vest"). Sumbu horizontal (x-axis) merepresentasikan jarak dalam satuan tertentu, mulai dari 2 hingga 12. Sementara itu, sumbu vertikal (y-axis) menggambarkan tingkat akurasi deteksi dalam persentase. Pada awal pengujian, jarak 2 meter menunjukkan akurasi sebesar 88%. Namun, ketika jarak ditingkatkan menjadi 3 meter, akurasi mengalami peningkatan signifikan hingga 90%. Akurasi tetap berada pada tingkat yang tinggi antara 86% pada jarak 4 hingga 6 meter. Ini menandakan bahwa dalam kisaran jarak ini, sistem deteksi bekerja optimal.

Namun pada jarak 7 meter, terjadi sedikit penurunan akurasi menjadi 82%. Penurunan lebih lanjut terlihat pada jarak 8 dan 9 meter dengan akurasi sebesar 76% dan 73%. Pada titik ini, sistem mulai menunjukkan penurunan performa yang lebih tajam. Dari jarak 10 meter akurasi menjadi 63%, jarak 11 meter akurasi mengalami penurunan yang signifikan 38% dan 37%, yang menandakan bahwa tidak maksimal

untuk deteksi. Penurunan performa sistem deteksi pada jarak yang lebih jauh disebabkan oleh beberapa faktor, salah satunya adalah penurunan kualitas gambar yang diambil oleh webcam atau perangkat pengambilan gambar lain karena kelebihan cahaya yang mana memantulkan cahaya lewat vest dan menjadikan kamera tidak dapat mendeteksi vest.


4.4.2 Pengujian Deteksi Secara Realtime Model YOLOV8 Posisi Dinamis

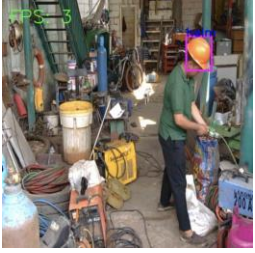
Pengujian model YoloV8 dalam kondisi langsung (real-time) di lingkungan kerja yang sesungguhnya. Pada pengujian ini, sistem deteksi berbasis YoloV8 digunakan untuk mendeteksi objek, seperti helm dan vest, ketika pekerja berada dalam posisi bebas atau acak di tempat kerja. Pengujian ini dilakukan untuk mengevaluasi performa model dalam kondisi nyata, di mana pekerja tidak berada pada posisi yang teratur atau statis, melainkan bergerak bebas di area kerja. Hal ini bertujuan untuk melihat seberapa baik YoloV8 dapat mendeteksi objek dalam situasi dinamis dan lingkungan yang tidak terkontrol.



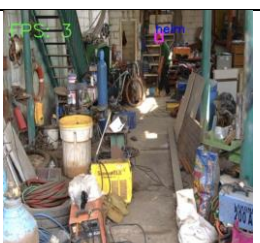

4.4.2.1 Pengujian Deteksi Secara Real-Time Pada Posisi Dinamis Dengan Variasi Jarak Untuk Helm

Pengujian deteksi secara real-time pada posisi dinamis dilakukan dengan memvariasikan jarak antara kamera dan subjek yang menggunakan helm antara 2 hingga 12 meter. Hasil pengujian dengan berbagai variasi jarak tersebut disajikan pada Tabel 4.7.

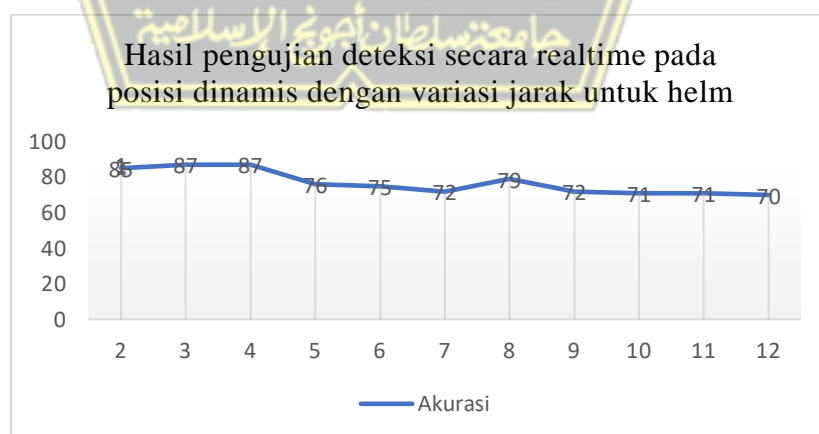
Tabel 4. 7 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk helm

No	Objek	Posisi	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Helm	Menghadap kanan dan jongkok	2	85	3	

2	Helm	Menghadap kanan	3	87	3	
3	Helm	Menghadap kanan dan jongkok	4	87	3	
4	Helm	Menghadap kanan dan membungkuk	5	76	3	
5	Helm	Menghadap kiri	6	75	3	
6	Helm	Menghadap kanan dan jongkok	7	72	3	
7	Helm	Menghadap kanan dan jongkok	8	79	3	

8	Helm	Membungkuk dan menghadap kiri	9	72	3	
9	Helm	Membelakangi kamera	10	71	3	
10	Helm	Membelakangi kamera	11	71	3	
11	Helm	Membungkuk	12	70	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.14



Gambar 4. 14 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk helm

Pada jarak 2 meter, akurasi deteksi mencapai 85% ketika subjek menghadap kanan dan jongkok, yang menunjukkan bahwa pada jarak dekat, meskipun posisi tidak ideal, sistem masih mampu mendeteksi dengan baik. Pada jarak 3 dan 4 meter, akurasi meningkat hingga 87% saat subjek tetap menghadap kanan, baik dalam posisi normal maupun jongkok. Hal ini mengindikasikan bahwa posisi subjek yang menghadap langsung ke kamera memberikan kualitas deteksi yang lebih tinggi karena cahaya yang terpantul dengan lebih optimal ke arah kamera.

Ketika pada jarak 5 meter, subjek menghadap kanan dan membungkuk, akurasi turun drastis menjadi 76%. Penurunan ini bisa disebabkan oleh sudut posisi subjek yang menyebabkan helm menjadi kurang terlihat atau sebagian terhalang. Pada jarak 6 hingga 7 meter, ketika subjek menghadap kiri atau dalam posisi jongkok, akurasi terus menurun hingga mencapai 72%. Perubahan arah menghadap dan posisi yang lebih dinamis dapat mengakibatkan memperburuk kemampuan sistem untuk mendeteksi helm dengan jelas, terutama ketika pencahayaan tidak memadai dan terlalu terang untuk menangkap detail helm dari sudut-sudut tersebut.


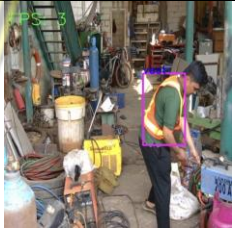


Pada jarak 8 meter, akurasi kembali meningkat hingga 79%, meskipun subjek tetap dalam posisi jongkok. Kenaikan ini bisa jadi disebabkan oleh penyesuaian pencahayaan atau sudut pandang yang lebih ideal pada saat pengambilan data. Namun, penurunan akurasi yang signifikan terjadi lagi pada jarak 9 hingga 12 meter, terutama ketika subjek membelakangi kamera atau membungkuk. Akurasi mencapai titik terendah 70% pada jarak 12 meter, yang menunjukkan bahwa jarak yang jauh, dikombinasikan dengan posisi subjek yang tidak ideal dan pencahayaan yang tidak optimal, sangat mempengaruhi performa deteksi sistem.



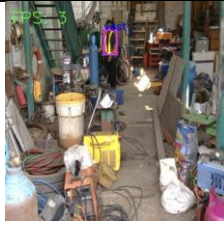


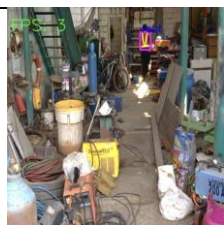

Secara keseluruhan, naik turunnya akurasi dalam tabel ini menunjukkan bahwa jarak, posisi subjek, dan pencahayaan memainkan peran penting dalam keakuratan deteksi helm. Posisi subjek yang menghadap kamera dengan baik dan pencahayaan yang memadai cenderung menghasilkan akurasi lebih tinggi, sedangkan sudut yang tidak menguntungkan dan jarak yang jauh mengurangi performa sistem secara signifikan.

4.4.2.2 Pengujian Deteksi Secara Real-Time Pada Posisi Dinamis Dengan Variasi Jarak Untuk Vest

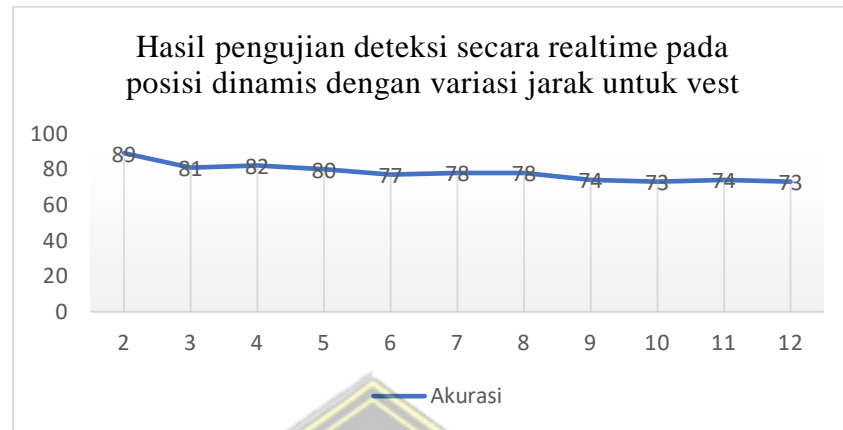
Pengujian deteksi secara real-time pada posisi dinamis dilakukan dengan memvariasikan jarak antara kamera dan subjek yang menggunakan vest. Jarak yang digunakan dalam pengujian ini berkisar antara 2 hingga 12 meter untuk mengevaluasi akurasi sistem dalam kondisi nyata. Hasil pengujian dengan berbagai variasi jarak tersebut disajikan pada Tabel 4.8.

Tabel 4. 8 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk vest

No	Objek	Posisi	Jarak (meter)	Akurasi (%)	FPS	Screenshot
1	Vest	Membelakangi kamera	2	89	3	
2	Vest	Menghadap kiri dan membungkuk	3	81	3	
3	Vest	Menghadap kiri dan membungkuk	4	82	3	
4	Vest	Menghadap kanan dan membungkuk	5	80	3	

5	Vest	Menghadap kiri dan jongkok	6	77	3	
6	Vest	Membelakangi kamera	7	78	3	
7	Vest	Menghadap kanan	8	78	3	
8	Vest	Menghadap kanan	9	74	3	
9	Vest	Menghadap kiri dan jongkok	10	73	3	
10	Vest	Membelakangi kamera	11	74	3	
11	Vest	Membungkuk dan Membelakangi kamera	12	73	3	

Dari hasil percobaan, diperoleh data yang ditampilkan pada Gambar 4.15



Gambar 4. 15 Hasil pengujian deteksi secara real-time pada posisi dinamis dengan variasi jarak untuk vest

Tabel tersebut menunjukkan fluktuasi akurasi deteksi objek vest pada berbagai posisi dan jarak, dengan faktor cahaya dan posisi subjek yang memengaruhi naik turunnya akurasi. Pada jarak 2 meter, akurasi tertinggi sebesar 89% terjadi ketika subjek membelakangi kamera, yang disebabkan oleh pencahayaan yang optimal dan posisi subjek yang stabil sehingga citra vest terdeteksi dengan jelas. Namun, ketika subjek menghadap kiri dan membungkuk pada jarak 3 meter, akurasi turun menjadi 81%. Posisi membungkuk dan arah menghadap yang tidak langsung ke kamera mempengaruhi penurunan ini, karena bagian vest tidak terdeteksi sepenuhnya oleh sistem.

Pada jarak 4 dan 5 meter, akurasi sedikit berfluktuasi antara 82% dan 80%, menunjukkan bahwa perubahan posisi subjek dari menghadap kiri ke kanan dan membungkuk tetap mempengaruhi kualitas deteksi, meskipun jaraknya bertambah. Pada jarak 6 meter, ketika subjek menghadap kiri dan jongkok, akurasi menurun lebih signifikan menjadi 77%, yang bisa disebabkan oleh sudut tubuh subjek yang lebih ekstrem dan pencahayaan yang tidak cukup optimal untuk menangkap citra dengan baik.

Penurunan akurasi berlanjut pada jarak 8 hingga 10 meter, di mana subjek

menghadap kanan atau jongsok, dengan akurasi berkisar antara 74% hingga 73%. Posisi ini dapat memperparah hilangnya detail visual dari vest karena semakin jauhnya jarak dan sudut pandang yang lebih tidak ideal. Pada jarak 11 dan 12 meter, akurasi tetap rendah di angka 74% dan 73%, terutama ketika subjek membungkuk dan membelakangi kamera, menandakan bahwa kombinasi posisi membelakangi, jarak yang jauh, serta kondisi pencahayaan yang kurang mendukung sangat memengaruhi turunnya performa deteksi. Akurasi deteksi sangat dipengaruhi oleh posisi subjek dan jarak, di mana posisi yang tidak langsung menghadap kamera dan jarak yang lebih jauh menurunkan kualitas deteksi, terutama ketika cahaya tidak cukup untuk memperjelas citra vest.

4.4.3 Analisis Pengujian Secara Realtime Model YOLOv8 pada Objek dalam Posisi Menghadap Kamera dan Posisi Dinamis

Pengujian real-time menggunakan model YOLOv8 bertujuan untuk menilai kinerja sistem dalam mendeteksi objek seperti helm dan rompi (vest) pada kondisi statis dan dinamis. Pada skenario pertama, pengujian dilakukan ketika subjek menghadap kamera dengan variasi jarak untuk melihat dampaknya terhadap akurasi deteksi, yang disajikan dalam bentuk tabel. Untuk deteksi helm, hasil menunjukkan akurasi tertinggi sebesar 89% pada jarak 3 meter, sementara akurasi terendah sebesar 73% terjadi pada jarak 12 meter. Penurunan akurasi secara bertahap setelah jarak 3 meter disebabkan oleh penurunan kualitas gambar yang diterima kamera dan kondisi pencahayaan yang kurang mendukung. Namun, penurunan ini tetap konsisten di kisaran 74% pada jarak 8 hingga 11 meter, sebelum turun lebih jauh pada jarak terjauh.

Sedangkan untuk deteksi vest, akurasi tertinggi juga tercapai pada jarak 3 meter dengan nilai 90%, namun penurunan akurasinya lebih signifikan dibandingkan helm. Pada jarak 10 meter, akurasi turun drastis hingga 52%, dan menurun lagi pada jarak 11 dan 12 meter dengan nilai 38% dan 37%. Penurunan tajam ini salah satu faktor dipengaruhi oleh pantulan cahaya pada vest yang mengurangi kemampuan sistem dalam mendeteksi objek pada jarak jauh. Pada pengujian posisi dinamis, di mana subjek bergerak atau berada dalam posisi tidak


statis seperti membungkuk atau membelakangi kamera, hasil deteksi lebih bervariasi. Akurasi deteksi helm tetap optimal pada jarak dekat dengan nilai 87% ketika subjek menghadap kamera atau sedikit membungkuk. Namun, akurasi menurun hingga 70% ketika subjek bergerak dinamis pada jarak 5 hingga 7 meter, menunjukkan bahwa posisi tubuh dan jarak mempengaruhi deteksi objek helm oleh sistem YOLOv8.




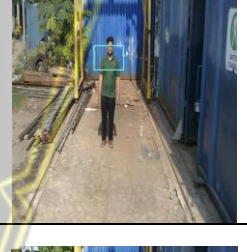


Deteksi vest pada posisi dinamis juga mengalami penurunan akurasi seiring bertambahnya jarak, terutama ketika subjek berada dalam posisi membelakangi kamera atau membungkuk. Pada jarak 2 hingga 4 meter, akurasi tetap tinggi antara 89% hingga 82%, namun turun drastis setelah jarak 8 meter. Secara keseluruhan, pengujian ini menunjukkan bahwa deteksi helm dan vest menggunakan YOLOv8 dipengaruhi oleh jarak, posisi dinamis, dan pencahayaan. Meskipun performa sistem baik pada jarak dekat dan posisi statis, kinerjanya menurun pada jarak jauh dan posisi dinamis. Ini mengindikasikan pentingnya pengaturan pencahayaan dan posisi kamera untuk mendukung kinerja deteksi yang optimal.





4.4.4 Pengujian Deteksi Haar Cascade Classifier

Selain melakukan pengujian terhadap model YoloV8 juga dilakukan pengujian terhadap model Haar cascade classifier. Pengujian ini dilakukan untuk mengetahui kinerja *Classifier* dalam melakukan deteksi pada wajah manusia. Deteksi ini dilakukan untuk memberitahu pada sistem bahwa ada pekerja di dalam frame. Deteksi yang dilakukan oleh Classifier hanya sebatas pada wajah manusia saja. Pada pengujian Haarcascade Classifier secara online ini menggunakan variasi jarak yang berbeda-beda yaitu 2 meter hingga 12 meter dengan input berupa kamera. dapat dilihat pada table 4.9 sebagai berikut :

Tabel 4. 9 Hasil pengujian model Haar cascade dengan variasi jarak

No	Jarak (meter)	Keterangan	<i>Screenshot</i>
1	2	Terdeteksi dengan sempurna	

2	3	Terdeteksi dengan sempurna	
3	4	Terdeteksi dengan sempurna	
4	5	Terdeteksi dengan sempurna	
5	6	Terdeteksi dengan sempurna	
6	7	Terdeteksi dengan sempurna	
7	8	Terdeteksi dengan sempurna	

8	9	Terdeteksi dengan sempurna	
9	10	Terdeteksi dengan sempurna	
10	11	Terdeteksi dengan sempurna	
11	12	Terdeteksi dengan sempurna	

Hasil pengujian menggunakan metode Haar Cascade untuk pengenalan wajah dan badan menunjukkan performa yang sangat baik dan stabil dalam mendeteksi objek pada berbagai jarak. Berdasarkan tabel, pengujian dilakukan dari jarak 2 meter hingga 12 meter, dan dalam semua pengujian, sistem mampu mendeteksi wajah dan badan secara sempurna.

Pada jarak 2 hingga 12 meter, tidak ada kegagalan deteksi yang tercatat. Hal ini menunjukkan bahwa metode Haar Cascade bekerja secara konsisten dalam mendeteksi objek, baik pada jarak dekat maupun jarak yang lebih jauh. Performanya stabil tanpa dipengaruhi oleh jarak dalam rentang yang diuji. Kemampuan deteksi yang sempurna pada jarak tersebut dapat disebabkan oleh kekuatan algoritma Haar Cascade yang menggunakan fitur cascade classifier untuk mendeteksi pola wajah dan tubuh manusia dengan efisien, bahkan dalam kondisi yang bervariasi. Dari hasil ini, dapat disimpulkan bahwa Haar Cascade adalah

metode yang andal untuk pengenalan wajah dan badan, setidaknya dalam jarak hingga 12 meter. Kestabilan deteksi yang sempurna juga menunjukkan bahwa faktor-faktor lain, seperti kualitas gambar dan kondisi lingkungan mempengaruhi deteksi.

4.5 Pengujian Keseluruhan

Pengujian keseluruhan sistem merupakan pengujian terakhir pada penelitian ini, bertujuan untuk mengevaluasi kinerja sistem deteksi perlengkapan kesehatan dan keselamatan kerja (K3) secara menyeluruh. Dalam pengujian ini, beberapa parameter terbaik yang telah diperoleh dari pengujian sebelumnya diterapkan untuk memastikan performa optimal dari model YOLO yang digunakan.

Kesuksesan pada pengujian ini ditentukan oleh kemampuan model untuk mendeteksi objek berupa helm dan vest dengan tingkat akurasi yang memenuhi standar yang telah ditetapkan. Selain itu, sistem juga harus mampu mengambil tangkapan layar (*screenshot*) sebagai bukti pelanggaran apabila pekerja tidak menggunakan perlengkapan K3 yang diwajibkan. Keberhasilan dalam aspek ini menjadi indikator bahwa sistem telah siap untuk diterapkan di lingkungan kerja nyata. Hasil pengujian keseluruhan dapat dilihat pada tabel 4.10.

Tabel 4. 10 Hasil Pengujian Keseluruhan

No	Objek	Jarak Uji (meter)	Posisi	Jarak Terbaik (meter)	FPS	Bukti pelanggaran
1	Manusia	2-12	Menghadap Kamera	3	3	Ya
2	Manusia	2-12	Dinamis	3	3	Ya
3	Hanya Helm	2-12	Menghadap Kamera	3	3	Ya
4	Hanya Helm	2-12	Dinamis	3	3	Ya
5	Hanya Rompi	2-12	Menghadap Kamera	3	3	Ya

6	Hanya Rompi	2-12	Dinamis	3	3	Ya
7	Helm dan rompi	2-12	Menghadap Kamera	3	3	Tidak
8	Helm dan rompi	2-12	Dinamis	3	3	Tidak

Dari percobaan yang telah dilakukan dan berdasarkan hasil pada tabel 4.10 sistem yang telah dibuat telah sesuai dengan kondisi yang diinginkan dimana pada saat kondisi objek perlengkapan keselamatan kerja tidak lengkap maka sistem akan melakukan penangkapan gambar dan disimpan pada memori dengan format nama waktu terjadinya pelanggaran.



Gambar 4. 16 Hasil Screen Capture Sebagai Bukti

Tabel yang disajikan menggambarkan hasil pengujian terbaik online/realtime serta offline menggunakan model YOLOv8 dalam mendeteksi keberadaan manusia, helm, dan rompi dalam berbagai kondisi di tempat kerja. Pengujian ini dilakukan dengan variasi jarak, posisi objek, dan kondisi dinamis atau statis, serta hasil pelanggaran yang terdeteksi. Hasil terbaik terdapat pada tabel 4.10 yang mana jarak terbaik adalah 3 meter. Meskipun hasil pada jarak lainnya juga menunjukkan performa yang baik, peneliti memilih jarak tersebut sebagai yang terbaik berdasarkan evaluasi keseluruhan pengujian.

Hasil pengujian yang disajikan dalam tabel menunjukkan bahwa sistem deteksi mampu mengenali pelanggaran penggunaan perlengkapan kesehatan dan

keselamatan kerja (K3) dengan tingkat akurasi terbaik pada jarak 3 meter, sistem berhasil mendeteksi pelanggaran, baik ketika objek berada dalam posisi menghadap kamera maupun dalam keadaan bergerak, terutama jika hanya helm atau rompi yang digunakan. Deteksi ini disertai dengan bukti berupa tangkapan layar sebagai dokumentasi pelanggaran. Hal serupa juga terlihat pada pengujian terhadap manusia tanpa perlengkapan lengkap, yang teridentifikasi sebagai pelanggaran.

Sebaliknya, jika pekerja mengenakan helm dan rompi sekaligus, baik dalam posisi menghadap kamera maupun dinamis, sistem tidak mencatat adanya pelanggaran berupa bukti screenshot. Namun perlu diperhatikan bahwa kecepatan sistem hanya mampu menghasilkan 3 FPS (frame per second), yang tergolong sangat rendah dibandingkan spesifikasi kamera yang mampu mencapai 30 FPS. Rendahnya FPS ini dapat dijelaskan oleh keterbatasan spesifikasi perangkat keras yang digunakan, seperti CPU atau GPU, yang mungkin tidak memadai atau belum dioptimalkan untuk menjalankan deteksi YOLO. Jika perangkat keras, seperti CPU atau GPU, yang digunakan untuk menjalankan YOLO tidak memiliki kekuatan yang cukup atau belum dioptimalkan, maka proses deteksi akan menjadi lebih lambat. Hal ini terutama terjadi saat menggunakan CPU, karena deteksi berbasis jaringan saraf membutuhkan banyak daya komputasi, yang berimbas pada penurunan FPS.

BAB V

PENUTUP

5.1 Kesimpulan

Perancangan sistem detektor dengan input awal berupa tangkapan citra dari kamera webcam Brio 100 yang dioperasikan melalui platform komputasi berbasis open-source, Visual Studio Code (VS Code). Objek yang terdeteksi diproses menggunakan metode Haar Cascade untuk mendeteksi keberadaan manusia. Apabila keberadaan manusia teridentifikasi, sistem melanjutkan ke tahap pendeteksian alat pelindung diri (APD) menggunakan algoritma YOLOv8. Pelanggaran yang terdeteksi kemudian dievaluasi menggunakan perhitungan mean Average Precision (mAP) yang diperoleh dari confusion matrix. Selanjutnya sistem secara otomatis mengklasifikasikan pelanggaran dan menghasilkan bukti dalam bentuk *screenshot*.

Pengaruh jarak antara kamera dengan objek memiliki dampak yang signifikan terhadap akurasi deteksi. Semakin jauh jarak objek dengan kamera, tingkat akurasi cenderung menurun. Hal ini disebabkan oleh ukuran objek yang terdeteksi menjadi semakin kecil, sehingga menyulitkan sistem untuk mengenali objek dengan tepat. Spesifikasi kamera turut memengaruhi kualitas citra yang dihasilkan. Jarak terbaik pengujian pada jarak 3 meter dengan tingkat akurasi sebesar 90% untuk deteksi rompi dan 89% untuk deteksi helm.

5.2 Saran

Pengumpulan dataset yang lebih besar, beragam dan sesuai sangat krusial untuk memberikan model pembelajaran referensi yang lebih luas dan mendalam, sehingga kemampuan model dalam mengenali pola serta situasi yang berbeda dapat ditingkatkan secara signifikan. Penggunaan perangkat dengan spesifikasi lebih tinggi juga menjadi aspek penting dalam mencapai kinerja deteksi yang optimal. Menambahkan perlengkapan keselamatan kerja lainnya, seperti masker, sarung tangan, dan sepatu pelindung.

REFERENSI

- [1] N. Haworth and S. Hughes, *The International Labour Organization*, Modul 5. Indonesia: SCORE, 2012. doi: 10.4337/9781849807692.00014.
- [2] A. F. Direja, Y. Cahyana, K. A. Baihaqi, I. Engineering, U. Buana, and P. Karawang, "Implementation Of The Yolov8 Method To Detect Work Safety," vol. 5, no. 3, pp. 865–871, 2024.
- [3] M. N. Nirvana, R. F. Rachmadi, and I. K. E. Purnama, "Sistem Pendeteksi Alat Pelindung Diri (APD) Pada Pekerja Konstruksi Berbasis Convolutional Neural Network," *J. Tek. ITS*, vol. 12, no. 3, pp. 1–7, 2023, doi: 10.12962/j23373539.v12i3.121850.
- [4] B. I. Arianto and E. Zuliarso, "Implementasi Metode YOLO pada Deteksi Pakaian Keselamatan yang Lengkap di Proyek Kontruksi," *J. Multidiscip. Res. Dev.*, vol. 6, no. 1, pp. 56–63, 2023, [Online]. Available: <https://creativecommons.org/licenses/by/4.0/>
- [5] M. A. Taufiqurrochman and H. Februariyanti, "Rancang Bangun Aplikasi Deteksi Alat Pelindung Diri (APD) untuk Pekerja Proyek dengan Menggunakan Algoritma Yolov5," *J. JTik (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 2, pp. 471–480, 2024, doi: 10.35870/jtik.v8i2.1960.
- [6] T. Informatika, F. Teknik, U. Nusantara, and P. Kediri, "Deteksi Kelalaian Alat Pelindung Diri Pada Pekerja Kontruksi Bangunan," *Semin. Nas. Inov. Teknol. Un Pgri Kediri*, vol. 6, pp. 58–63, 2022.
- [7] 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.
- [8] M. I. Dinata, N. Sulistianingsih, and S. A. A. Yusuf, "Implementasi Deep

Learning Dalam Klasifikasi Citra Gambar Dengan Menggunakan Metode CNN,” *J. Inf. Technol. Syst.*, pp. 14–19, 2023.

- [9] S. Yulina, “Penerapan Haar Cascade Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV,” *J. Komput. Terap.*, vol. 7, no. 1, pp. 100–109, 2021, doi: 10.35143/jkt.v7i1.3411.
- [10] S. K. Shandilya, A. Srivastav, K. Yemets, A. Datta, and A. K. Nagar, “YOLO-based segmented dataset for drone vs. bird detection for deep and machine learning algorithms,” *Data Br.*, vol. 50, p. 109355, 2023, doi: 10.1016/j.dib.2023.109355.
- [11] F. Saputra and M. Rizky Mahaputra, “Building Occupational Safety and Health (K3): Analysis of the Work Environment and Work Discipline,” *J. Law, Polit. Humanit.*, vol. 2, no. 3, pp. 105–114, 2022, doi: 10.38035/jlph.v2i3.91.
- [12] A. Purwanto *et al.*, “Peningkatan Keselamatan Kerja Melalui Pelatihan ISO 45001:2018 Sistem Manajemen Keselamatan dan Kesehatan Kerja Pada Industri Manufaktur di Tangerang,” *J. Community Serv. Engagem.*, vol. 01, no. 02, pp. 1–6, 2021, [Online]. Available: <https://jocosae.org/index.php/jocosae/article/view/8>
- [13] N. Nafi’iyah, “Algoritma Kohonen dalam Mengubah Citra Graylevel Menjadi Citra Biner,” *J. Ilm. Teknol. Inf. Asia*, vol. 9, no. 2, pp. 49–55, 2015, [Online]. Available: <https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/125>
- [14] A. Sujjada and E. Juniar, “Implementasi Algoritma Hill Cipher Untuk Proses Enkripsi Data Menggunakan Media Citra Digital,” *Ris. Tek. Inform. dan Komput.*, vol. 3, no. 1, pp. 1–17, 2021, [Online]. Available: <https://restikom.nusaputra.ac.id>
- [15] Muhammad Nur Ihsan Muhlashin and A. Stefanie, “Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8,” *JATI (Jurnal*

Mhs. Tek. Inform., vol. 7, no. 2, pp. 1363–1368, 2023, doi: 10.36040/jati.v7i2.6927.

- [16] F. J. Du and S. J. Jiao, “Improvement of Lightweight Convolutional Neural Network Model Based on YOLO Algorithm and Its Research in Pavement Defect Detection,” *Sensors*, vol. 22, no. 9, 2022, doi: 10.3390/s22093537.
- [17] J. Tursunov, G. Memonova, and M. Gulnoza, “Custom Object Detection Using Yolo,” *9th Int. Sci. Pract. Conf. "SCIENCE Pract. Implement. TO Mod. Soc.*, pp. 392–402, 2021.
- [18] G. I. E. Soen, M. Marlina, and R. Renny, “Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants,” *JITU J. Inform. Technol. Commun.*, vol. 6, no. 1, pp. 24–30, 2022, doi: 10.36596/jitu.v6i1.781.

