
Implementasi Arsitektur Yolo V8 Dalam Mendeteksi Alat Pelindung Diri (APD) Di Sektor Konstruksi Dan Industri

Jeicman Samperante¹⁾, I Made Agus Wirahadi Putra²⁾, Putu Adi Guna Permana³⁾
Sistem Informasi¹⁾³⁾, Manajemen Informatika²⁾
Institut Teknologi dan Bisnis STIKOM Bali
Denpasar, Indonesia

e-mail: 210030462@stikom-bali.ac.id¹⁾, wirahadi@stikom-bali.ac.id²⁾,
putuadi_guna@stikom-bali.ac.id³⁾

Abstrak

Kecelakaan kerja di sektor konstruksi dan industri terus meningkat, dengan salah satu penyebab utamanya adalah kelalaian dalam penggunaan alat pelindung diri (APD), seperti helm keselamatan. Pemantauan manual yang dilakukan oleh petugas sering kali kurang efektif dan rentan terhadap kesalahan manusia. Untuk mengatasi masalah ini, penelitian ini mengembangkan sistem deteksi otomatis penggunaan helm dengan algoritma YOLOv8, yang dilatih menggunakan dataset dari Roboflow di platform Google Colab. Hasil pelatihan model kemudian diimplementasikan dalam aplikasi Streamlit untuk pemantauan real-time. Pengujian menggunakan metode black box testing menunjukkan bahwa semua skenario berhasil dijalankan, dan model mencapai nilai mean Average Precision (mAP50) sebesar 0.779, menunjukkan akurasi tinggi dalam berbagai kondisi. Dengan hasil ini, sistem yang dikembangkan dapat menjadi solusi efektif dalam meningkatkan keselamatan kerja di lingkungan konstruksi dan industri.

Kata kunci: YOLOv8, deteksi helm, keselamatan kerja, computer vision, Streamlit.

1. Pendahuluan

Jumlah kecelakaan kerja di Indonesia dalam tiga tahun terakhir terus meningkat. Hingga akhir tahun 2022, jumlah kecelakaan kerja yang tercatat mencapai 265.334 kasus. Jumlah ini meningkat dari 234.370 kasus pada tahun 2021, dan 221.740 kasus pada tahun 2020 [1]. Salah satu sektor dengan risiko tinggi adalah sektor konstruksi, di mana menurut laporan dari International Labour Organization (ILO), satu dari enam kecelakaan fatal terjadi di sektor ini [2].

Beberapa penyebab kecelakaan dengan tingkat kejadian yang tinggi sebenarnya dapat dikurangi melalui penggunaan APD, seperti helm keselamatan, vest, sepatu, sarung tangan dan kaca mata pelindung. Oleh sebab itu, pengawasan terhadap penggunaan APD, menjadi aspek yang sangat penting untuk dilakukan.

Untuk memantau penggunaan alat pelindung diri (APD) oleh karyawan, beberapa perusahaan telah menyediakan departemen pengawasan khusus. Namun, hal ini dinilai tidak cukup aman karena adanya kesalahan manusia yang dapat memengaruhi proses pemantauan alat pelindung diri. Berdasarkan permasalahan tersebut, diperlukan suatu perangkat untuk memonitor penggunaan alat pelindung diri secara independen dan memiliki masa pakai yang lama.

Salah satu solusi untuk membangun sistem pemantauan alat pelindung diri adalah dengan menerapkan teknologi computer vision. Dengan penerapan computer vision, proses pengawasan dapat dilakukan oleh kamera dengan melibatkan kemampuan deteksi objek. Penelitian sebelumnya terkait dengan deteksi alat pelindung diri pernah dilakukan dengan beberapa metode, yaitu RPA-SSD, YOLO, dan CNN.

Penelitian dengan metode CNN pernah dilakukan, namun pada penelitian tersebut deteksi alat pelindung diri dilakukan dengan pendekatan klasifikasi, sehingga sistem tidak mampu mendeteksi dengan baik apabila di dalam satu citra terdapat lebih dari satu orang yang menggunakan APD yang berbeda [3]. Sistem deteksi APD menggunakan metode CNN juga pernah dilakukan dengan pendekatan object detection, namun hasil performa belum maksimal karena sistem kesulitan dalam mendeteksi objek seperti sarung tangan. Hal ini dipengaruhi oleh pergerakan manusia pada saat proses deteksi yang menyebabkan objek sulit tertangkap dengan baik oleh kamera [4].

Sementara itu, penelitian untuk melakukan deteksi objek dengan metode RPA-SSD menyebutkan bahwa sistem masih memerlukan pengembangan untuk dapat meningkatkan kemampuan deteksi terhadap objek dengan skala kecil [5]. Selain RPA-SSD, YOLO V3 juga pernah digunakan, di mana penelitian ini berfokus untuk menguji tiga pendekatan berbeda dengan YOLO V3. Hasilnya, penelitian ini dinilai kurang maksimal karena model kesulitan untuk mendeteksi objek kecil [6].

Deteksi APD juga pernah dilakukan dengan YOLO V5, di mana penelitian tersebut berfokus pada modifikasi YOLO V5 untuk meningkatkan performa YOLO V5 terkait objek yang gagal terdeteksi atau kesalahan deteksi yang disebabkan oleh permasalahan deteksi objek kecil dan skala objek. YOLO V5s tanpa modifikasi hanya menghasilkan MAP 85,9% [7].

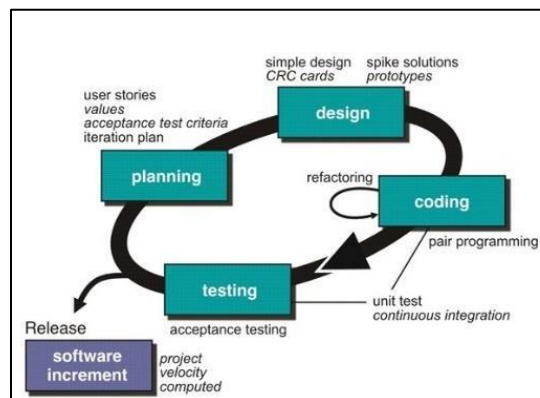
Berdasarkan hasil dari penelitian-penelitian sebelumnya, dapat dilihat bahwa terdapat dua kekurangan utama pada sistem deteksi APD yang telah dikembangkan. Kekurangan pertama adalah sistem yang belum mampu melakukan deteksi dengan baik jika pada satu citra terdapat dua atau lebih orang yang menggunakan APD berbeda. Kekurangan kedua adalah model yang kesulitan untuk mendeteksi objek berukuran kecil.

Pada penelitian ini akan dibangun sistem dengan algoritma You Only Look Once (YOLO V8) untuk mendeteksi penggunaan APD seperti helm keselamatan secara real-time. Sistem ini dirancang untuk mendeteksi objek secara bersamaan dalam satu frame, bahkan jika terdapat beberapa individu yang menggunakan APD berbeda. Penelitian ini juga mengembangkan dataset yang lebih banyak dan variatif serta menggunakan fitur pelabelan bounding box untuk meningkatkan akurasi deteksi, serta melakukan konfigurasi hyperparameter agar mendapatkan hasil optimal.

2. Metode Penelitian

2.1 Metode Pengembangan Sistem

Metode pengembangan perangkat sistem adalah proses pengorganisasian sekumpulan metode dan konvensi notasi yang telah didefinisikan untuk mengembangkan perangkat lunak atau sistem yang akan dibangun[8]. Dalam penelitian ini metode pengembangan sistem yang digunakan adalah metode Extreme Programming (XP), yang terdiri dari empat tahapan utama seperti yang digambarkan pada Gambar 2, yaitu: Planning, Design, Coding, dan Testing [9], [10].



Gambar 1. Metode Waterfall

- Planning**, Tahap planning dalam penelitian ini mencakup analisis kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi kemampuan sistem menerima input berupa gambar, video, dan tangkapan layar, menampilkan bounding box pada objek terdeteksi, serta menyajikan hasil deteksi dan klasifikasi sesuai kelas.
- Desain**, merupakan gambaran visual pada program, disini peneliti akan melakukan perancangan desain pada sistem yang akan dibuat untuk kemudian diimplementasikan [11]. Tahapan pada desain sistem ini meliputi, perancangan diagram dan perancangan interface.
- Coding**, tahap dimana dilakukannya pengerjaan atau pembuatan sistem. Pada tahap ini peneliti mengimplementasikan program menggunakan *coding* (bahasa komputer) sesuai dengan desain yang sudah dibuat sebelumnya.
- Testing**, Dalam penelitian ini, tahap pengujian menggunakan metode black box testing. Metode ini dipilih karena fokus utamanya adalah menguji fungsionalitas aplikasi tanpa memerlukan pemahaman tentang kode atau struktur internal sistem. Proses pengujian dilakukan dengan memberikan input kepada aplikasi dan memeriksa output yang dihasilkan. Pengujian ini bertujuan untuk memastikan bahwa aplikasi beroperasi sesuai dengan spesifikasi yang telah dirancang.

3. Hasil dan Pembahasan

3.1 Planning

Pada tahap planning, dilakukan analisis kebutuhan dan perancangan konsep aplikasi. Analisis kebutuhan ini dibagi menjadi dua kategori, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

a. Kebutuhan Fungsional

Berikut adalah beberapa kebutuhan fungsional untuk sistem yang akan dibangun dalam penelitian ini:

1. Sistem dapat menerima inputan gambar, video dan tangkapan layar.
2. Sistem dapat menampilkan bounding box objek yang dideteksi.
3. Sistem dapat menampilkan hasil output deteksi serta klasifikasi sesuai dengan kelas.

b. Kebutuhan Non Fungsional

Berikut adalah beberapa kebutuhan Non-fungsional untuk sistem yang akan dibangun dalam penelitian ini:

1. Perangkat Keras(Hardware)

Penelitian ini menggunakan spesifikasi perangkat keras (hardware) dengan rincian seperti yang tercantum pada tabel 1 berikut.

Tabel 1. Spesifikasi Hardware

No	Perangkat Keras	Keterangan
1	Processor	Intel Core i5-12500H
2	Ram	8GB
3	Storage	512 GB SSD
4	Grapichs Card	NVIDIA GeForce RTX 3050
5	Monitor	15,6 Inch Full HD
6	Resolusi	144Hz

2. Perangkat Lunak(Software)

Penelitian ini menggunakan spesifikasi perangkat lunak (software) dengan rincian seperti yang tercantum pada Tabel 2 berikut.

Tabel 2. Spesifikasi Hardware

No	Perangkat Keras	Keterangan
1	Processor	Intel Core i5-12500H
2	Ram	8GB
3	Storage	512 GB SSD
4	Grapichs Card	NVIDIA GeForce RTX 3050
5	Monitor	15,6 Inch Full HD
6	Resolusi	144Hz

3.2 Design

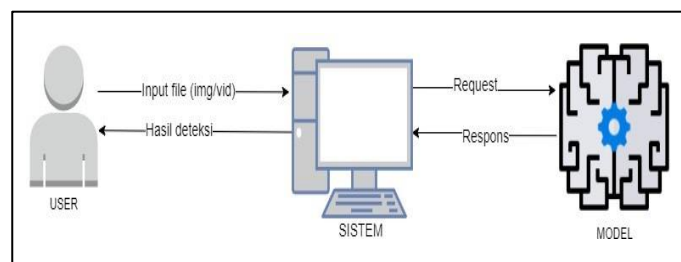
Tahapan desain dalam penelitian berfokus pada perancangan diagram dan alur sistem yang akan mendukung dalam pengembangan sistem.

a. Perancangan Diagram

Perancangan diagram mencakup diagram arsitektur sistem, pembuatan alur metodologi, diagram aktivitas (activity diagram), dan diagram use case, yang semuanya bertujuan untuk menggambarkan alur kerja dan interaksi antar komponen dalam sistem.

1. Diagram Arsitektur Sistem

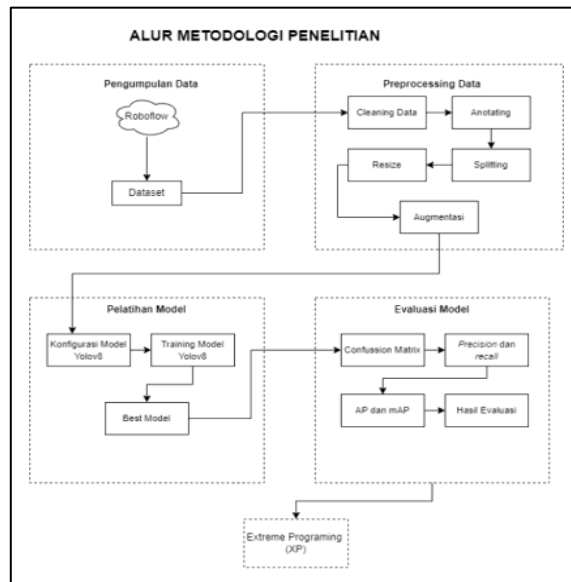
Arsitektur sistem dirancang untuk memberikan gambaran mengenai cara kerja sistem, proses yang dijalankan, serta output data yang dihasilkan. Gambar 2 menunjukkan arsitektur sistem yang digunakan dalam penelitian ini.



Gambar 2 Arsitektur Sistem Diagram

2. Alur Metodologi

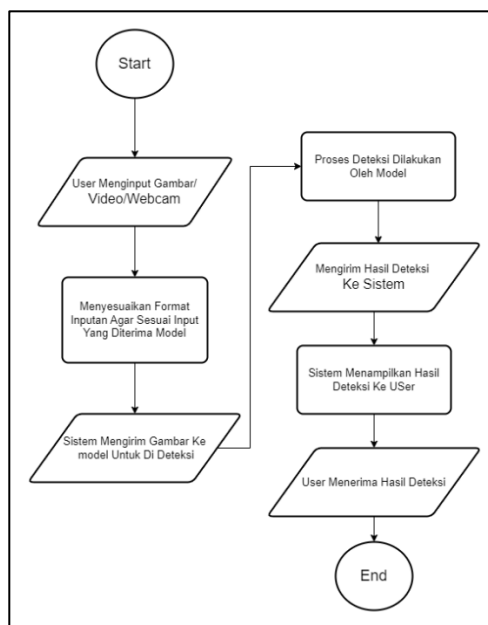
Perancangan alur metodologi bertujuan untuk menggambarkan tahapan tahapan selama proses pengerjaan model yang akan dibangun. Tahapan pada alur metodologi ini terdiri dari 4 tahapan utama yaitu Pengumpulan Data , Data Preprocessing , Pelatihan Model , dan Evaluasi model



Gambar 3 Alur Metodologi Penelitian

3. Flowchart Diagram

Diagram ini digunakan untuk menggambarkan alur yang terjadi dalam proses deteksi. Gambar 4 menunjukkan flowchart diagram yang digunakan dalam penelitian ini.



Gambar 4 Flowchart Diagram

3.3 Coding

Tahap coding merupakan langkah di mana desain yang telah direncanakan sebelumnya diterjemahkan menjadi kode program yang dapat dijalankan, menggunakan bahasa pemrograman Python. Pada penelitian ini, model dilatih menggunakan Google Colab untuk memanfaatkan sumber daya komputasi berbasis cloud yang efisien. Hasil dari proses pelatihan ini kemudian digunakan dalam aplikasi Streamlit untuk memfasilitasi visualisasi dan interaksi pengguna dengan model yang telah dikembangkan. Berikut adalah hasil dari proses coding :



Gambar 5 Hasil Coding Berupa Streamlit

3.4 Testing

Pada tahap testing model yang sudah jadi akan diuji coba menggunakan metode blackbox testing untuk menguji fungsionalitas aplikasi yang sudah dibangun. Kemudian testing juga akan mengevaluasi hasil dari performa yang telah dilatih sebelumnya. Berikut tabel pengujian menggunakan metode blackbox :

Tabel 3 Pengujian Blackbox

No	Test Case	Hasil yang diharapkan	Hasil
1	Pengguna dapat mengakses perangkat lunak	Pengguna berhasil mengakses perangkat lunak	Berhasil
2	Pengguna dapat menginput file gambar	Pengguna berhasil menginputkan file gambar	Berhasil
3	Pengguna dapat menginput file video	Pengguna berhasil menginputkan file video	Berhasil
4.	Pengguna dapat mengakses screen live deteksi	Pengguna berhasil mengakses screen live	Berhasil
5	Sistem dapat menampilkan hasil deteksi dari file gambar	Sistem berhasil menampilkan hasil deteksi dari file gambar	Berhasil
6	Sistem dapat menampilkan hasil deteksi dari file video	Sistem berhasil menampilkan hasil deteksi dari file video	Berhasil
7	Sistem dapat menampilkan hasil deteksi dari screen live	Sistem berhasil menampilkan hasil deteksi dari screen live	Berhasil

Untuk hasil evaluasi model, model dievaluasi setiap kelas untuk melihat performa pada beberapa paramater seperti Precision Recall MAP50 dan mAP50-95 berikut adalah hasil evaluasi :

Class	Images	Instances	Box(P	R	mAP50	mAP50-95) :
all	812	3067	0.741	0.768	0.784	0.48
Goggles	70	90	0.704	0.633	0.661	0.289
boots	413	856	0.725	0.84	0.854	0.572
gloves	195	345	0.543	0.47	0.513	0.247
helmet	615	1002	0.886	0.948	0.941	0.568
vest	548	774	0.847	0.948	0.953	0.723

Gambar 6 Hasil Evaluasi

Hasil akhir dari proses pelatihan model menunjukkan bahwa model yang telah dilatih mencapai nilai mean Average Precision (mAP50) sebesar 0.779, yang mencerminkan tingkat akurasi sekitar 77.9% dalam mendeteksi dan mengklasifikasikan objek sesuai dengan target yang ditetapkan. Nilai mAP berkisar antara 0 hingga 1, di mana 1 menunjukkan deteksi sempurna dan 0 berarti deteksi tidak akurat. Dengan nilai 0.779, model ini menunjukkan kinerja yang cukup baik, terutama dalam deteksi objek yang kompleks, dan berada dalam kisaran nilai mAP yang umumnya dianggap baik, yaitu 0.7 hingga 0.8.

4. Kesimpulan

Implementasi algoritma YOLOv8 dalam mendeteksi helm sebagai alat pelindung di sektor konstruksi dan industri terbukti memberikan hasil yang akurat dan efisien, dengan kemampuan deteksi real-time yang dapat mengidentifikasi helm secara otomatis pada berbagai kondisi pencahayaan dan sudut pandang. Pelatihan model dilakukan di Google Colab menggunakan dataset dari Roboflow, memungkinkan eksperimen berbasis cloud yang efisien dan fleksibel, sementara hasil model diimplementasikan dalam aplikasi Streamlit untuk deteksi real-time yang mudah diakses pengguna. Pengujian dengan metode black box testing menunjukkan bahwa semua skenario berhasil dijalankan dengan baik, dan model mampu mendeteksi helm dengan akurasi tinggi, mencapai nilai mean Average Precision (mAP50) sebesar 0.779. Meskipun nilai ini menunjukkan performa yang solid dalam mendukung sistem pemantauan keselamatan kerja secara otomatis, masih ada ruang untuk peningkatan, terutama jika dibandingkan dengan nilai mAP yang lebih tinggi, seperti 0.9 atau lebih, yang umumnya diinginkan dalam aplikasi deteksi objek yang lebih kritis.

Daftar Pustaka

- [1] Jenderal Pembinaan Pengawasan Ketenagakerjaan, "Laporan Kecelakaan Kerja di Indonesia" 2022.
 - [2] International Labour Organization (ILO), *Safety and Health in Construction*. ILO, 2015.
 - [3] M. Rifki and others, "Penerapan Convolutional Neural Network untuk Deteksi Alat Pelindung Diri pada Sektor Konstruksi," *Jurnal Teknologi dan Sistem Komputer*, vol. 18, no. 2, pp. 133–139, 2019.
 - [4] Z. Zainudin and others, "Deteksi Objek Menggunakan CNN pada Penggunaan Alat Pelindung Diri di Sektor Konstruksi," *Jurnal Teknik Elektro dan Komputer*, vol. 9, no. 4, pp. 125–132, 2019.
 - [5] J. Wu and others, "Pengembangan RPA-SSD untuk Deteksi Objek Berukuran Kecil dalam Pengawasan APD," *Jurnal Pengolahan Citra dan Komputer*, vol. 24, no. 1, pp. 52–61, 2019.
 - [6] A. Nath and others, "Evaluasi YOLO V3 untuk Deteksi Objek APD," *Jurnal Teknologi dan Inovasi*, vol. 19, no. 3, pp. 98–106, 2020.
 - [7] L. Han and J. Zeng, "Optimasi YOLO V5 untuk Deteksi APD dalam Lingkungan Kerja," *Jurnal Pengolahan Citra*, vol. 30, no. 1, pp. 23–30, 2022.
 - [8] P. Adi and G. Permana, "Scrum Method Implementation in a Software Development Project Management," 2015. [Online]. Available: www.ijacsa.thesai.org
 - [9] S. Widiastuti, N. Hendrastuty, N. Penulis Korespondensi, and S. Widiastuti Submitted, "Rancang Bangun Sistem Informasi Kepegawaian Berbasis Website Dengan Menggunakan Metode Extreme Programming Pada Kantor Kelurahan Komerling Agung Kecamatan Gunung Sugih," vol. x, pp. 291–301, 2023, doi: 10.33365/jtsi.
 - [10] T. Ardiansah, "Perancangan Sistem Persediaan Menggunakan Metode Extreme Programming," *Jurnal Ilmiah Informatika Dan Ilmu Komputer (JIMA-ILKOM)*, vol. 1, no. 1, pp. 1–6, 2022.
 - [11] C. Binardo, "Pengembangan Sistem Pendaftaran Kejuaraan Karate Berbasis Web dengan Pendekatan Extreme Programming," *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 2, no. 2, pp. 276–284, 2021.
-