

Sistem Deep-Learning Yolov8 untuk Deteksi Penggunaan APD Secara Real-Time

Nelson Mandela Rande Langi ^{(1)*}, Arif Fadillullah ⁽²⁾

Departemen Teknik Komputer, Universitas Borneo Tarakan, Tarakan, Indonesia

e-mail : usernelson90@gmail.com, arif.fadl@borneo.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 28 Februari 2025, direvisi 17 September 2025, diterima 23 September 2025, dan dipublikasikan 25 Januari 2026.

Abstract

Although workplace safety regulations in construction are clear, many workers are still reluctant to use Personal Protective Equipment (PPE) due to a lack of awareness, work pressure, and limited facilities. As a result, the risk of serious accidents increases. Conventional approaches such as verbal warnings or CCTV monitoring are considered less effective for early detection and prevention of violations. This study proposes an automatic detection system for PPE usage in construction areas using YOLOv8. The model was trained on a secondary dataset of 3,569 images for 100 epochs, with a 60% training, 20% validation, and 20% test split. Testing on 90 real-time frames showed good performance in detecting 8 PPE classes, with an average precision of 0.935, recall of 0.806, and F1-measure of 0.862. The results indicate that the system can classify PPE usage with high accuracy. However, a recall below 1 suggests that some objects, particularly "not wearing glasses" and "not wearing shoes," failed to be detected. The F1-measure of 0.862 reflects a good balance between precision and recall.

Keywords: Personal Protective Equipment, Computer Vision, Deep Learning, YOLOv8, Object Detection

Abstrak

Meskipun aturan keselamatan kerja di konstruksi sudah jelas, banyak pekerja masih enggan menggunakan APD (Alat Pelindung Diri) karena kurangnya kesadaran, tekanan kerja, dan minimnya fasilitas. Akibatnya, risiko kecelakaan serius meningkat. Pendekatan konvensional seperti teguran lisan atau pemantauan CCTV dinilai kurang efektif untuk deteksi dini dan pencegahan pelanggaran. Penelitian ini mengusulkan sistem deteksi otomatis penggunaan APD di area konstruksi menggunakan YOLOv8. Model dilatih dengan 3.569 dataset sekunder selama 100 epoch, dengan pembagian data 60% training, 20% validasi, dan 20% testing. Pengujian pada 90 frame *real-time* menunjukkan kinerja baik dalam mendeteksi 8 kelas APD, dengan nilai masing-masing rata-rata precision 0,935, recall 0,806, dan F1-measure 0,862. Hasil menunjukkan sistem mampu mengklasifikasikan penggunaan APD dengan akurasi baik. Namun, recall di bawah 1 mengindikasikan beberapa objek, terutama "tidak pakai kacamata" dan "tidak pakai sepatu," gagal terdeteksi. Nilai F1-measure 0,862 mencerminkan keseimbangan precision dan recall yang baik.

Kata Kunci: Alat Pelindung Diri, Computer Vision, Deep Learning, YOLOv8, Deteksi Objek

1. PENDAHULUAN

Penggunaan APD merupakan salah satu implementasi sistem manajemen keselamatan dan kesehatan kerja. menurut Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia Nomor PER.08/MEN/VII/2010 Tentang Alat Pelindung Diri tahun 2010 adalah suatu alat yang biasa digunakan untuk melindungi seseorang atau pekerja dari potensi bahaya di lingkungan kerja. Meskipun ada peraturan dan pedoman yang mengatur peraturan K3 di area konstruksi, masih saja sering terjadi pelanggaran yang mengakibatkan risiko yang tidak perlu bagi para pekerja. Pelanggaran ini dapat disebabkan oleh berbagai faktor, termasuk tekanan waktu, kurangnya pengawasan, dan kurangnya konsekuensi yang tegas terhadap pelanggaran. Pada tahun 2019, sektor industri pengolahan mencatat jumlah hari kerja hilang tertinggi, yaitu 87.599 hari, dengan 10.872 kasus pekerja sementara tidak dapat bekerja. Dalam dunia konstruksi, 29%



Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

kecelakaan kerja disebabkan oleh kejatuhan benda, 26% akibat tergelincir atau terpukul. Hingga tahun 2020, jumlah kecelakaan kerja meningkat hingga 177.000 kasus (Alfidyani et al., 2020). Khusus untuk di wilayah provinsi Kaltara (Kalimantan Utara) sendiri, berdasarkan catatan dari Badan Pusat Statistika (BPS) Kaltara, tahun 2022 ditemukan angka kecelakaan kerja sebanyak 670 kasus dan pada tahun 2023 meningkat sebanyak 685 kasus. Artinya terdapat penambahan 15 kasus kecelakaan kerja yang terjadi di Kaltara dari tahun 2022 sampai dengan 2023 (Julianti & Wibawa, 2024). Meskipun ada peraturan dan pedoman yang mengatur K3 dan penggunaan APD di area kontruksi, masih seringkali terjadi pelanggaran yang mengakibatkan risiko yang fatal bagi para pekerja. Pelanggaran ini dapat disebabkan oleh berbagai faktor, termasuk tekanan waktu, kurangnya pengawasan, dan kurangnya konsekuensi yang tegas terhadap pelanggaran K3 (Wulandari, 2023).

Penelitian serupa mengenai pendekripsi APD secara otomatis sudah pernah dilakukan sebelumnya, seperti penelitian dengan judul “*Deep learning-Based Automatic Safety Helmet Detection System for Construction Safety*” yang diteliti oleh Hayat & Morgado-Dias (Hayat & Morgado-Dias, 2022) yang menggunakan Kumpulan data *benchmark* yang berisi 5.000 gambar helm digunakan dalam penelitian ini, yang dibagi lagi dengan perbandingan 60:20:20 (%) untuk pelatihan, pengujian, dan validasi, masing-masing. Hasil eksperimen menunjukkan bahwa arsitektur YOLO tercapai rata-rata precision (mAP) terbaik sebesar 92,44%, sehingga menunjukkan hasil yang sangat baik dalam pendekripsi helm pengaman bahkan dalam kondisi cahaya redup. Akan tetapi penelitian tersebut masih menggunakan satu jenis dataset APD yaitu helm safety dan masih belum mencakup ke semua jenis APD yang digunakan pada area kontruksi seperti rompi, sepatu, kacamata, dan helm safety. Mengacu pada permasalahan yang telah diidentifikasi, penelitian ini mengusulkan sebuah sistem deteksi *real-time* menggunakan teknik deep learning, khususnya YOLOv8, untuk mendekripsi penggunaan APD secara multi-objek pada area konstruksi. Sistem ini mengintegrasikan kamera sensor dan algoritma YOLOv8 untuk secara akurat mengidentifikasi pekerja yang menggunakan atau tidak menggunakan APD.

2. METODE PENELITIAN

Penelitian ini mengoptimalkan pengembangan teknologi *Computer Vision* yang mengkombinasikan penggunaan kamera sensor, sistem ini juga menggunakan Deep-learning YOLOv8 (*You Only Look Once*) sebagai algoritma pemrosesan deteksi multi objek manusia yang menggunakan/tidak menggunakan APD. Kelebihan utama *Deep-learning* YOLO adalah kecepatan deteksi yang tinggi tanpa mengorbankan akurasi yang signifikan. Metode ini mampu melakukan deteksi multi objek secara *Real-time* dengan kinerja yang baik, sehingga cocok untuk berbagai aplikasi yang membutuhkan respons cepat seperti kendaraan otonom pengawasan keamanan, dan pengenalan wajah (Alfarizi et al., 2023).

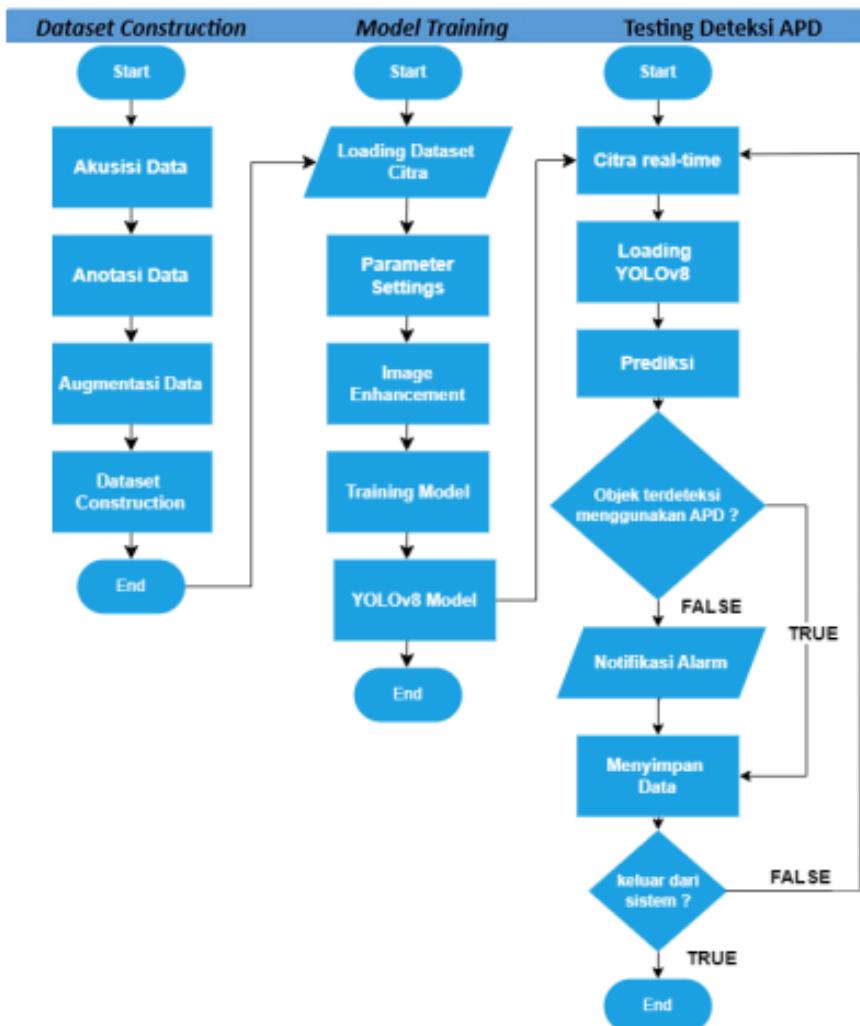
2.1 Perancangan Sistem

Perancangan sistem dapat dialurkan seperti pada flowchart pada Gambar 1. Tahapan dimulai dari *dataset construction* sampai pada testing deteksi APD. *Dataset construction* merupakan langkah awal yang sangat krusial dalam pengembangan model pembelajaran mesin berbasis citra. Kualitas dan kuantitas *dataset* yang baik akan sangat mempengaruhi kinerja model yang dihasilkan. Proses ini melibatkan beberapa tahapan penting, yaitu: (1) pengumpulan data, yakni mengumpulkan citra yang relevan, bervariasi, dan berkualitas; (2) pre-processing data, seperti pembersihan, augmentasi, dan normalisasi untuk meningkatkan kualitas citra; (3) pelabelan data, yakni memberikan label pada objek atau kelas yang terdapat dalam citra; (4) pembagian data menjadi *training set*, *validation set*, dan *test set*; serta (5) penyimpanan data dengan format dan struktur yang efisien.

Visi komputer menggunakan subset dari instans data, yang dikenal sebagai data pelatihan, untuk menyesuaikan parameter suatu model. Subset lain dari data dikenal sebagai data pengujian, digunakan untuk mengevaluasi model dan memperkirakan kemampuan model untuk menggeneralisasi ke gambar yang tidak terlihat selama pelatihan (Scheuerman et al., 2021). Penelitian ini akan mengembangkan sistem deteksi otomatis APD pada area konstruksi



menggunakan algoritma Deep-learning YOLOv8 yang dilatih pada dataset citra berjumlah 3.400 citra yang akan diberi label. Dataset tersebut akan dibagi menjadi tiga kelompok, yaitu data latih (60%), data validasi (20%), dan data uji (20%) untuk meningkatkan akurasi dan generalisasi model (Hayat & Morgado-Dias, 2022).



Gambar 1 Flowchart Sistem

Tahapan model training adalah proses pelatihan kumpulan dataset citra, sehingga model dapat menemukan pola data dan memprediksi data baru. Dalam proses ini terdapat 3 tahapan utama yang akan dilakukan sistem, yaitu *image enhancement*, *parameter settings*, dan *model training*. Pelatihan model dalam visi komputer adalah proses mengajarkan komputer untuk 'memahami' gambar. Proses ini dimulai dengan mempersiapkan kumpulan data gambar yang akan digunakan untuk melatih model. Gambar-gambar ini kemudian diolah untuk meningkatkan kualitasnya agar fitur-fitur penting lebih terlihat. Setelah itu, model akan diatur parameternya, seperti jumlah lapisan dan tingkat pembelajaran. Tahap akhir adalah proses pelatihan itu sendiri, di mana model akan mempelajari pola-pola dalam data gambar secara berulang-ulang. Semakin banyak data yang digunakan dan semakin baik pengaturan parameternya, maka model akan semakin akurat dalam memprediksi atau mengklasifikasikan objek dalam gambar baru (Santos et al., 2022). Pada tahap *testing*, sistem mengambil frame video *real-time* dari lingkungan baru. Model YOLOv8 menganalisis objek dan memberi label sesuai penggunaan APD. Jika terdeteksi tidak memakai APD, sistem segera mengaktifkan alarm peringatan.



2.2 Teknik Pengumpulan Data

Penelitian ini bertujuan mengembangkan model deteksi objek *real-time* berbasis deep learning YOLOv8 untuk mendekripsi penggunaan APD pada area konstruksi. Dataset yang terdiri dari 3.400 citra berlabel akan digunakan untuk melatih, memvalidasi, dan menguji model. Pembagian dataset ini mengikuti proporsi umum dalam penelitian deep learning, yaitu 60% untuk pelatihan, 20% untuk validasi, dan 20% untuk pengujian. Setelah model dianggap memiliki akurasi yang memadai, model akan dievaluasi lebih lanjut menggunakan data sampel *real-time*. Pengumpulan data sampel akan dilakukan terhadap 3, 5, dan 7 sampel citra *real-time* objek (random menggunakan APD/tidak) sekaligus dalam 1 kali pengambilan data yang diulang tiap 30 detik sebanyak 30 kali pada frame dalam waktu tertentu secara random, sehingga akan diperoleh 90 set hasil data uji sistem terhadap precision, recall, dan F1-measure beserta rata-rata akhir dari ketiga jenis pengujian terhadap seluruh data hasil uji sistem (Hand et al., 2021).

2.3 Teknik Analisis Data

Pengujian riset akan dihitung berdasarkan metrik precision, recall, dan F1-measure. Precision, dalam konteks evaluasi model klasifikasi, adalah metrik yang mengukur proporsi dari prediksi positif yang benar terhadap total prediksi positif. Secara matematis, precision didefinisikan sebagai rasio antara *true positive* dan jumlah dari *true positive* dan *false positive* (Prabowo, 2021). Recall adalah metrik yang mengukur proporsi dari contoh positif yang benar-benar diklasifikasikan sebagai positif terhadap total jumlah contoh positif yang sebenarnya ada. Secara matematis, recall didefinisikan sebagai rasio antara *true positive* dan jumlah dari *true positive* dan *false negative* (Wijaya et al., 2022). F1-measure adalah metrik evaluasi yang menggabungkan precision dan recall. Ini memberikan kita gambaran yang lebih komprehensif tentang kinerja model kita, terutama ketika kita ingin menyeimbangkan antara kemampuan model untuk memberikan prediksi positif yang benar (precision) dan kemampuan model untuk menemukan semua contoh positif yang sebenarnya ada (recall) (Rahma et al., 2021).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1measure = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

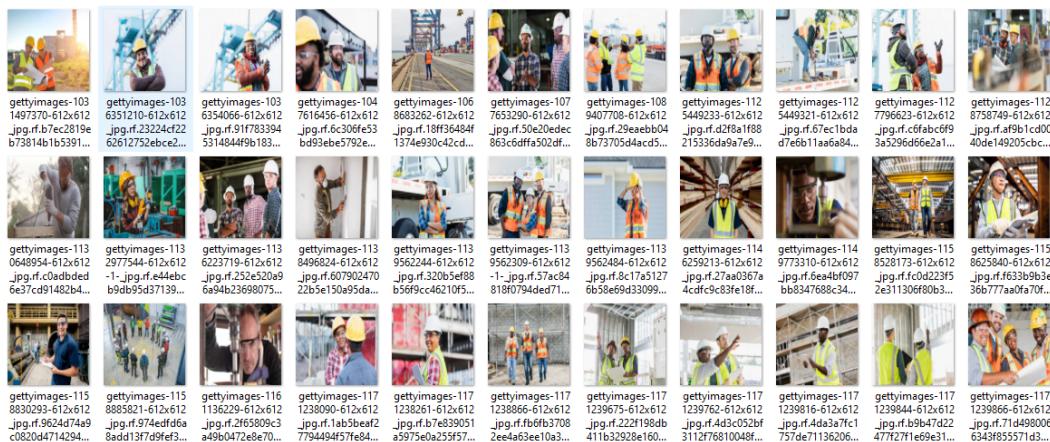
Cara penafsiran dan penyimpulan kebenaran hasil prediksi sistem usulan ini akan dihitung berdasarkan 3 metrik perhitungan dapat dilihat pada Pers. (1) untuk perhitungan precision, recall pada Pers. (2), dan F1-measure pada Pers. (3), yang akan digunakan untuk menghitung hasil dari data uji. Nilai-nilai tersebut berdasarkan hasil identifikasi dari: a) TP (True Positive), yaitu jumlah objek sebenarnya yang terdeteksi benar oleh sistem; b) FP (False Positive), yaitu jumlah objek noise yang terdeteksi benar oleh sistem; c) FN (False Negative), yaitu jumlah objek sebenarnya yang tidak terdeteksi benar oleh sistem (Surbakti et al., 2021).

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Dataset

Dataset yang digunakan dalam pelatihan model berjumlah 1.740 data citra. Dataset ini diambil dari sumber pustaka https://bit.ly/CHVG_Dataset, yang menyediakan kumpulan data penggunaan APD yang relevan dan sesuai dengan kebutuhan penelitian. Contoh data dalam dataset tersebut dapat dilihat pada Gambar 2, yang menunjukkan variasi penggunaan APD di lingkungan konstruksi.

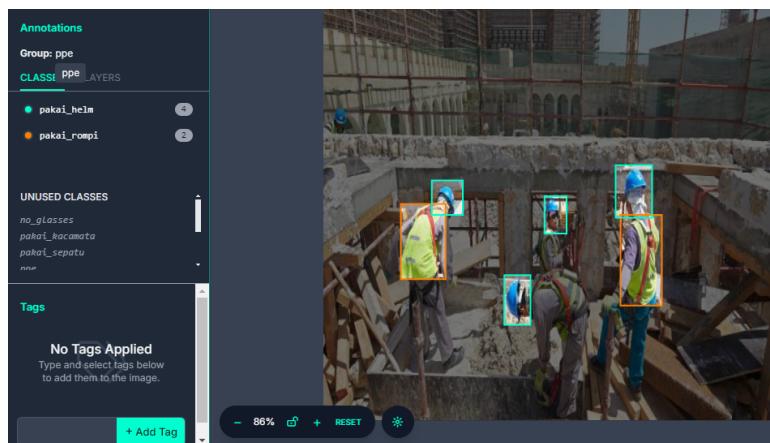




Gambar 2 Kumpulan Dataset

3.2 Anotasi Data

Anotasi dimulai dengan menentukan kelas-kelas objek yang akan dideteksi. Penentuan kelas dilakukan secara jelas agar model dapat mengidentifikasi objek dengan akurat. Gambar 3 memperlihatkan tahap awal dalam pembangunan model deteksi objek, yaitu proses anotasi data. Pada tahap ini, tujuan utama yang ingin dicapai adalah memberikan label yang sesuai dengan kelas yang ingin dideteksi sehingga model dapat belajar membedakan perbedaan antar objek (Supriyanto & Siahaan, 2024).



Gambar 3 Proses Anotasi Data

Tabel 1 Index Kelas

Index	Class
0	pakai_helm
1	pakai_kacamata
2	pakai_rompi
3	pakai_sepatu
4	tidak_pakai_helm
5	tidak_pakai_kacamata
6	tidak_pakai_rompi
7	tidak_pakai_sepatu

Proses anotasi dilakukan untuk seluruh kelas objek yang tercantum pada Tabel 1, dengan total 8 kelas. Setiap kelas diberikan indeks unik yang dimulai dari 0 untuk memudahkan proses

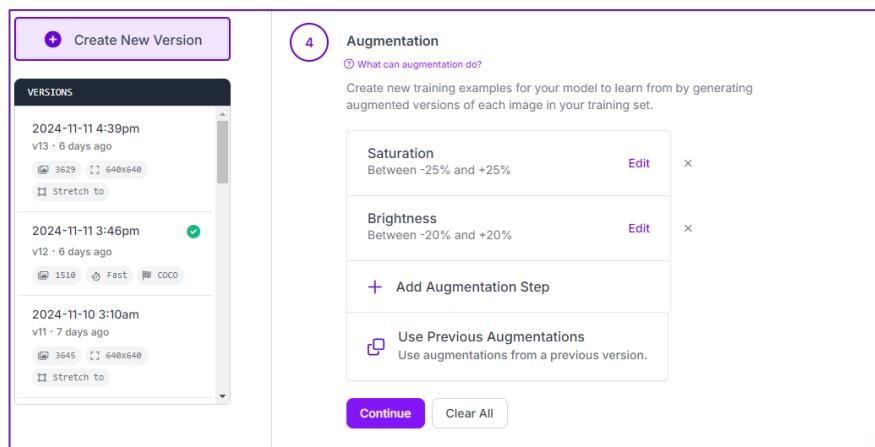


Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

identifikasi dan klasifikasi oleh model. Proses pelabelan menghasilkan informasi berupa koordinat objek dalam bentuk *bounding box* beserta nama kelasnya. Nilai koordinat dinyatakan dalam skala 0 hingga 1. Struktur data yang dihasilkan terdiri dari lima bagian, yaitu: kolom pertama berisi indeks kelas objek; kolom kedua dan ketiga menunjukkan posisi sumbu x dan y dari titik tengah *bounding box*; kolom keempat menunjukkan lebar; dan kolom kelima menunjukkan tinggi *bounding box* (Yasen et al., 2023). Informasi indeks dan nama kelas inilah yang nantinya akan ditampilkan pada *bounding box* ketika objek berhasil terdeteksi sesuai kelasnya masing-masing.

3.3 Augmentasi Data

Untuk meningkatkan kualitas serta kuantitas data pelatihan, dilakukan proses augmentasi pada 1.500 citra awal. Augmentasi ini dilakukan dengan memanipulasi tingkat kecerahan (*brightness*) dan saturasi warna (*saturation*) pada citra, sehingga menghasilkan 3.569 citra baru yang memiliki variasi pencahayaan dan warna yang lebih beragam. Variasi tersebut diharapkan dapat meningkatkan kemampuan model dalam mengenali objek dalam berbagai kondisi pencahayaan (Ganda & Bunyamin, 2021). Proses penambahan augmentasi melalui platform Roboflow ditampilkan pada Gambar 4.



Gambar 4 Proses Augmentasi Data

3.4 Training Data

Setelah proses pengolahan data selesai, langkah selanjutnya adalah melakukan pelatihan data. Proses pelatihan dilakukan menggunakan Google Colab, sebuah platform yang menyediakan akses gratis ke GPU berbasis cloud dengan durasi hingga dua belas jam per hari. Penggunaan GPU sangat membantu karena perangkat ini dirancang untuk melakukan komputasi paralel dalam jumlah besar, sehingga secara signifikan mempercepat proses pelatihan model *machine learning* (Arip et al., 2024).

Model	size (pixels)	mAP _{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Gambar 5 Varian Model YOLOv8



Pada tahap pelatihan dataset, model YOLO mulai mempelajari seluruh dataset yang telah melalui proses prapengolahan. Dalam penelitian ini digunakan model YOLOv8m (versi medium), yang merupakan salah satu dari lima varian YOLOv8 seperti ditunjukkan pada Gambar 5. Setiap varian memiliki karakteristik berbeda terkait ukuran model, kecepatan deteksi, serta akurasi, sehingga pemilihan varian disesuaikan dengan kebutuhan sistem. Perbandingan karakteristik antarvarian dapat dilihat pada Tabel 2.

Setelah memastikan data siap, langkah berikutnya adalah menginstal modul Python Ultralytics, yaitu modul resmi yang dikembangkan oleh tim pembuat algoritma YOLO. Modul ini menyediakan berbagai fungsi yang dibutuhkan untuk pelatihan dan pengujian model deteksi objek. Instalasi dilakukan melalui perintah *pip install ultralytics* pada terminal atau *command prompt*, seperti ditunjukkan pada Gambar 6.

```
Downloading ultralytics_thop-2.0.11-py3-none-any.whl (26 kB)
Installing collected packages: ultralytics-thop, ultralytics
  Successfully installed ultralytics-8.3.29 ultralytics-thop-2.0.11
```

Gambar 6 Instalasi Modul Ultralytics

Proses pelatihan model dilakukan dengan mengulang pembelajaran terhadap dataset sebanyak 100 kali, yang disebut dengan 100 epoch. Pada setiap epoch, model menyesuaikan parameter internalnya agar semakin akurat dalam mendeteksi objek pada citra. *Script* untuk menjalankan proses pelatihan ditunjukkan pada Gambar 7, sedangkan hasil pelatihan selama 100 epoch ditampilkan pada Tabel 3, yang memuat informasi mengenai perkembangan performa model sepanjang proses pelatihan.

```
import os

from ultralytics import YOLO

# Load a model
model = YOLO("yolov8m.yaml") # build a new model from scratch

# Use the model
results = model.train(data=os.path.join(ROOT_DIR, "config.yaml"), epochs=100,
                      loss_weights=[1.5, 0.3, 0.2]) # train the model
```

Gambar 7 Script Pelatihan Data

Tabel 2 Hasil Iterasi 100 Epoch

Epoch	box_loss	cls_loss	dfl_loss
10	2,135	2,598	2,261
20	1,873	2,004	1,95
30	1,785	1,811	1,871
40	1,694	1,623	1,79
50	1,629	1,464	1,699
60	1,58	1,381	1,665
70	1,504	1,275	1,603
80	1,446	1,209	1,574
90	1,391	1,121	1,515
100	1,286	0,9271	1,481

3.5 Validasi Data

Validasi data dilakukan untuk menghindari kesalahan interpretasi serta hasil analisis yang bias akibat penggunaan data yang tidak valid. Melalui proses validasi, hasil dari proses *training* dapat



Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

dievaluasi kembali sebelum dibentuk menjadi model akhir. Gambar 8 menunjukkan hasil validasi data yang dijalankan setelah proses *training* selesai. Pada tahap ini, dataset validasi memiliki peran krusial sebagai tolok ukur independen untuk mengevaluasi kinerja model setelah melalui proses pembelajaran. Dataset ini dipisahkan secara eksplisit dari dataset pelatihan dan digunakan untuk menguji kemampuan generalisasi model terhadap data yang belum pernah ditemui sebelumnya, sehingga dapat mengurangi risiko *overfitting*. Visualisasi dan informasi hasil validasi secara lebih rinci ditampilkan pada Tabel 4.

```
Validating runs/detect/train/weights/best.pt...
WARNING ! validating an untrained model YAML will result in 0 mAP.
Ultralytics 8.3.29 Python-3.10.12 torch-2.5.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8m summary (fused): 218 layers, 25,844,392 parameters, 0 gradients, 78.7 GFLOPs
  Class   Images  Instances   Box(P)      R      mAP50  mAP50-95: 100%|██████████
    all     2117     7928     0.769     0.83     0.889     0.565
    pakai_helm    588     1187     0.812     0.953     0.969     0.668
    pakai_kacamata   428      629     0.729     0.722     0.842     0.487
    pakai_rompi    382      760     0.81      0.949     0.974     0.686
    pakai_sepatu    326      751     0.82      0.883     0.920     0.605
    tidak_pakai_helm 359     1539     0.711     0.873     0.897     0.548
    tidak_pakai_kacamata 352     1313     0.675     0.624     0.727     0.366
    tidak_pakai_rompi 446     1254     0.764     0.891     0.945     0.602
    tidak_pakai_sepatu 361      595     0.831     0.746     0.836     0.555
Speed: 0.3ms preprocess, 9.8ms inference, 0.0ms loss, 2.5ms postprocess per image
Results saved to runs/detect/train
```

Gambar 8 Hasil Validasi Data

Tabel 3 Hasil Validasi Data

Class	images	instances	mAP
All	2117	7928	0,889
pakai_helm	588	1187	0,969
pakai_kacamata	428	629	0,842
pakai_rompi	382	760	0,974
pakai_sepatu	326	751	0,92
tidak_pakai_helm	359	1539	0,897
tidak_pakai_kacamata	352	1313	0,727
tidak_pakai_rompi	446	1254	0,945
tidak_pakai_sepatu	361	595	0,836

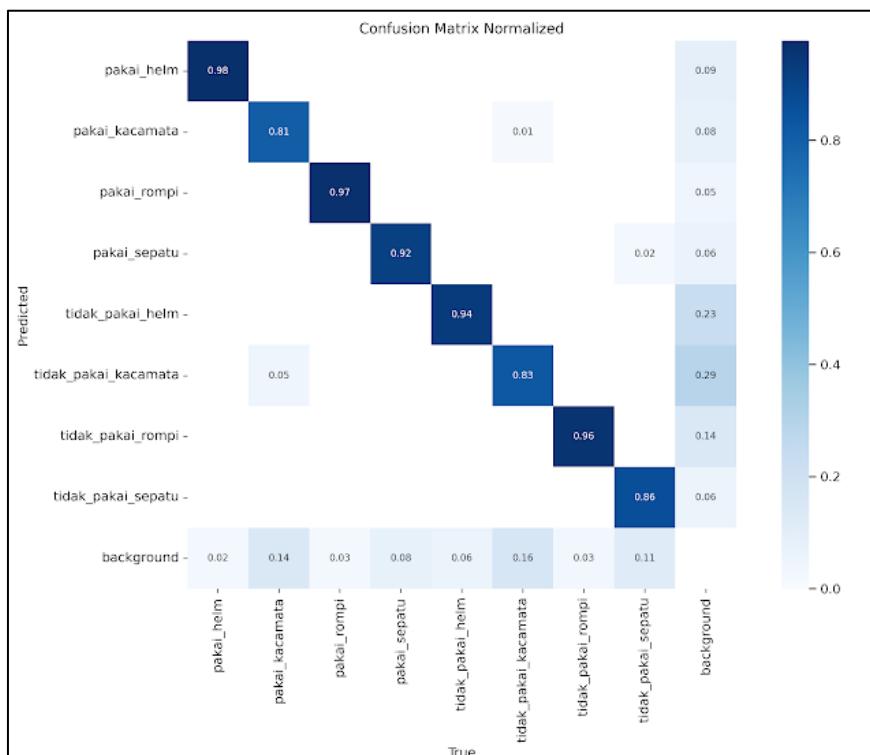
3.6 Evaluasi Hasil Model YOLOv8

Evaluasi model YOLOv8 merupakan langkah penting untuk memastikan bahwa model yang dibangun dapat diandalkan dan diterapkan dalam kondisi nyata. Melalui proses evaluasi, peneliti dapat memahami kinerja model, mengidentifikasi kelemahan, serta mengambil langkah yang diperlukan untuk meningkatkan performa agar sesuai dengan kebutuhan aplikasi (Muhlashin & Stefanie, 2023). Salah satu metode evaluasi yang digunakan adalah *confusion matrix*, yaitu matriks yang menggambarkan kinerja model pada data uji dengan membandingkan hasil prediksi model dengan label sebenarnya (Normawati & Prayogi, 2021). *Confusion matrix* dari hasil pelatihan model divisualisasikan pada Gambar 7. Tahap *testing* ini dilakukan setelah seluruh proses *training* dan *validation* selesai, dengan tujuan utama mengukur kemampuan model dalam mengenali serta mengklasifikasikan objek yang belum pernah ditemui sebelumnya. Dengan demikian, data *testing* berfungsi sebagai tolok ukur independen untuk mengevaluasi kemampuan generalisasi model terhadap data baru (Azhar et al., 2021).

Untuk memberikan gambaran yang lebih jelas, nilai numerik dari *confusion matrix* pada Gambar 9 ditampilkan dalam Tabel 5. Berdasarkan hasil pada tabel tersebut, evaluasi model YOLOv8 menunjukkan adanya variasi kinerja pada masing-masing kelas. Kelas dengan performa sangat baik meliputi *pakai_helm*, *pakai_rompi*, dan *tidak_pakai_rompi*, ditandai dengan nilai *precision*,



recall, dan F1-measure yang tinggi serta jumlah kesalahan deteksi yang rendah. Kelas *pakai_sepatu* dan *tidak_pakai_helm* memiliki performa moderat, dengan nilai precision yang perlu ditingkatkan karena jumlah *false positive* (FP) yang cukup besar, meskipun nilai recall-nya relatif baik. Sementara itu, beberapa kelas seperti *pakai_kacamata*, *tidak_pakai_kacamata*, dan *tidak_pakai_sepatu* menunjukkan performa yang kurang optimal. Rendahnya nilai precision pada kelas-kelas tersebut menunjukkan bahwa model sering melakukan kesalahan deteksi, ditambah jumlah *false negative* (FN) yang besar mengindikasikan banyak sampel relevan tidak terdeteksi. Secara khusus, kelas *tidak_pakai_kacamata* memiliki nilai precision terendah, yaitu 0,597, sehingga menjadi fokus penting untuk peningkatan performa pada penelitian selanjutnya.



Gambar 9 Confusion Matrix

Tabel 4 Nilai Metrik Hasil Evaluasi Pelatihan

Nama Kelas	Precision	Recall	F1-measure
pakai_helm	0,839	0,978	0,903
pakai_kacamata	0,709	0,808	0,755
pakai_rompi	0,865	0,972	0,916
pakai_sepatu	0,731	0,919	0,814
tidak_pakai_helm	0,719	0,936	0,813
tidak_pakai_kacamata	0,597	0,826	0,693
tidak_pakai_rompi	0,773	0,960	0,856
tidak_pakai_sepatu	0,639	0,864	0,735

3.7 Pengujian Sistem Deteksi APD

Untuk mengevaluasi kinerja model deteksi APD pada skenario nyata, digunakan dataset uji berupa 90 frame hasil deteksi real-time yang diambil di area konstruksi gedung Fakultas Kedokteran Universitas Borneo Tarakan. Dataset ini mencakup variasi jumlah objek, kelengkapan APD, serta berbagai kondisi lingkungan yang menantang, seperti pencahayaan yang acak, jarak pengamatan berkisar 3–5 meter, dan pergerakan objek (pekerja). Gambar 10 menampilkan contoh frame dari data uji yang digunakan.





Gambar 10 Frame Data Uji

3.8 Analisis Hasil Data Uji

Meskipun model YOLOv8 telah mencapai rata-rata precision sebesar 0,935, recall sebesar 0,806, dan F1-measure sebesar 0,862 dalam mendekripsi delapan kelas APD sebagaimana ditunjukkan pada Tabel 6, masih terdapat ruang untuk peningkatan, terutama pada nilai recall. Nilai precision yang tinggi mengindikasikan bahwa ketika model memprediksi suatu objek sebagai APD, prediksinya benar sekitar 93,5% dari waktu. Namun, nilai recall sebesar 0,806 menunjukkan bahwa model hanya mampu mendekripsi sekitar 80,6% dari seluruh penggunaan APD yang terdapat dalam data uji. Kombinasi kedua metrik tersebut menghasilkan nilai F1-measure sebesar 0,862, yang menggambarkan kinerja keseluruhan yang baik. Untuk memperoleh sistem yang lebih optimal dalam menangkap seluruh kasus positif, peningkatan nilai recall menjadi fokus yang penting.

Secara keseluruhan, dengan rata-rata precision 0,935, recall 0,806, dan F1-measure 0,862, hasil evaluasi menunjukkan bahwa sistem memiliki kinerja yang sangat baik dalam mendekripsi penggunaan APD multiobjek secara *real-time*. Nilai precision yang tinggi meminimalkan kemungkinan terjadinya *false positive* (deteksi objek yang sebenarnya tidak ada), sedangkan nilai recall menunjukkan kemampuan model dalam meminimalkan *false negative* (kegagalan mendekripsi objek yang sebenarnya ada). Sementara itu, nilai F1-measure sebagai rata-rata harmonis dari precision dan recall menunjukkan adanya keseimbangan yang baik antara kedua metrik tersebut, mengindikasikan kinerja sistem yang optimal secara keseluruhan.

Tabel 5 Hasil Evaluasi 90 Data Uji

Nama Kelas	Precision	Recall	F1-measure
pakai_helm	1	0,834	0,909
pakai_rompi	1	0,897	0,946
pakai_kacamata	0,811	0,785	0,798
pakai_sepatu	0,89	0,782	0,833
tidak_pakai_helm	0,984	0,891	0,935
tidak_pakai_rompi	0,924	0,965	0,944
tidak_pakai_kacamata	0,922	0,673	0,778
tidak_pakai_sepatu	0,948	0,619	0,749
Overall	0,935	0,806	0,862



4. KESIMPULAN

Model YOLOv8 dilatih menggunakan 3.569 data dengan 100 epoch dan pembagian 60% untuk training, 20% validasi, serta 20% testing. Evaluasi pada 20% data testing menunjukkan kelas pakai_helm, pakai_rompi, dan tidak_pakai_rompi memiliki performa optimal dengan precision, recall, dan F1-measure tinggi serta tingkat kesalahan deteksi rendah. Kelas pakai_sepatu dan tidak_pakai_helm menunjukkan recall tinggi tetapi precision rendah akibat tingginya false positive. Sebaliknya, kelas pakai_kacamata, tidak_pakai_kacamata, dan tidak_pakai_sepatu memiliki performa suboptimal, dengan precision rendah dan false negative tinggi.

Pada pengujian dengan 90 frame dari data video *real-time* yang diambil secara primer, model YOLOv8 telah menunjukkan performa yang baik dalam mendeteksi 8 kelas penggunaan APD. Dengan nilai rata-rata precision 0,935, model ini cukup akurat dalam mendeteksi objek yang terdeteksi sebagai APD. Namun, nilai rata-rata recall yang masih di bawah 1 (0,806) mengindikasikan bahwa model masih melewatkannya beberapa deteksi kelas pada model, terutama pada kelas tidak_pakai_kacamata dan tidak_pakai_sepatu. Nilai rata-rata F1-measure sebesar 0,862 menunjukkan bahwa model mencapai keseimbangan yang baik antara kemampuan untuk mengklasifikasikan objek dengan benar (precision) dan kemampuan untuk menemukan semua objek yang ada (recall).

Pengujian model YOLOv8 dengan menggunakan data primer sebaiknya dilakukan dengan jumlah data yang lebih banyak dan bervariasi. Data tersebut perlu mencakup berbagai skenario penggunaan Alat Pelindung Diri (APD), baik dari segi jenis maupun kondisi pemakaiannya, serta dalam beragam posisi tubuh, misalnya posisi duduk, berdiri, berjalan, maupun aktivitas lain yang relevan. Hal ini penting agar data uji memiliki karakteristik yang sebanding dengan data latih yang digunakan pada proses pengembangan model YOLOv8.

Dalam penelitian ini porsi data latih mencapai sekitar 20% dari total keseluruhan data (1.740 gambar), yaitu kurang lebih 348 gambar. Oleh karena itu, jumlah data primer yang digunakan untuk pengujian tidak ideal apabila hanya terbatas pada 90 gambar. Jumlah tersebut sebaiknya ditingkatkan agar minimal setara dengan jumlah data uji pada dataset sekunder, sehingga hasil evaluasi model menjadi lebih valid, representatif, dan dapat dipertanggungjawabkan secara ilmiah.

DAFTAR PUSTAKA

- Alfarizi, D. N., Pangestu, R. A., Aditya, D., Setiawan, M. A., & Rosyani, P. (2023). Penggunaan Metode YOLO pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis. *AI dan SPK: Jurnal Artificial Intelligent dan Sistem Penunjang*, 1(1), 54–63. <https://jurnalmahasiswa.com/index.php/aidanspk/article/view/144>
- Alfidyani, K. S., Lestantyo, D., & Wahyuni, I. (2020). Hubungan Pelatihan K3, Penggunaan APD, Pemasangan Safety Sign, dan Penerapan SOP dengan Terjadinya Risiko Kecelakaan Kerja (Studi pada Industri Garmen Kota Semarang). *Jurnal Kesehatan Masyarakat*, 8(4), 478–483. <https://doi.org/10.14710/jkm.v8i4.27531>
- Arip, A. A. S., Sazali, N., Kadirgama, K., Jamaludin, A. S., Turan, F. M., & Razak, N. Ab. (2024). Object Detection for Safety Attire Using YOLO (You Only Look Once). *Journal of Advanced Research in Applied Mechanics*, 113(1), 37–51. <https://doi.org/10.37934/aram.113.1.3751>
- Azhar, K. M., Santoso, I., & Soetrisno, Y. A. A. (2021). Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Algoritma YOLO dalam Sistem Pendekripsi Uang Kertas Rupiah Bagi Penyandang Low Vision. *Transient: Jurnal Ilmiah Teknik Elektro*, 10(3), 502–509. <https://doi.org/10.14710/transient.v10i3.502-509>
- Ganda, L. H., & Bunyamin, H. (2021). Penggunaan Augmentasi Data pada Klasifikasi Jenis Kanker Payudara dengan Model ResNet-34. *Jurnal Strategi*, 3(1), 187–193. <https://strategi.it.maranatha.edu/index.php/strategi/article/view/246>
- Hand, D. J., Christen, P., & Kirielle, N. (2021). F*: An Interpretable Transformation of the F-Measure. *Machine Learning*, 110(3), 451–456. <https://doi.org/10.1007/s10994-021-05964-1>



- Hayat, A., & Morgado-Dias, F. (2022). Deep Learning-Based Automatic Safety Helmet Detection System for Construction Safety. *Applied Sciences*, 12(16), Article ID: 8268. <https://doi.org/10.3390/app12168268>
- Julianti, I., & Wibawa, Y. (2024). Angka Kecelakaan Kerja Naik 15 Kasus. Benuanta. <https://benuanta.co.id/index.php/2024/02/07/angka-kecelakaan-kerja-naik-15-kasus/134245/18/24/08/>
- Muhlashin, M. N. I., & Stefanie, A. (2023). Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(2), 1363–1368. <https://doi.org/10.36040/jati.v7i2.6927>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier dan Confusion Matrix pada Analisis Sentimen Berbasis Teks pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2), 697–711. <https://doi.org/10.30645/J-SAKTI.V5I2.369>
- Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia Nomor PER.08/MEN/VII/2010 Tentang Alat Pelindung Diri, Kemenakertrans RI (2010). <https://jdih.kemnaker.go.id/peraturan/detail/158/peraturan-menteri-nomor-8-tahun-2010>
- Prabowo, T. T. (2021). Efektivitas Sistem Temu Kembali Informasi Perpustakaan Digital Institut Seni Indonesia (ISI) Yogyakarta dalam Tinjauan Recall dan Precision. *Media Pustakawan*, 28(1), 37–48. <https://doi.org/10.37014/medpus.v28i1.1087>
- Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, 2(3), 213–232. <https://doi.org/10.47747/jurnalknik.v2i3.534>
- Santos, C., Aguiar, M., Welfer, D., & Belloni, B. (2022). A New Approach for Detecting Fundus Lesions Using Image Processing and Deep Neural Network Architecture Based on YOLO Model. *Sensors*, 22(17), Article ID: 6441. <https://doi.org/10.3390/s22176441>
- Scheuerman, M. K., Hanna, A., & Denton, R. (2021). Do Datasets Have Politics? Disciplinary Values in Computer Vision Dataset Development. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2), 1–37. <https://doi.org/10.1145/3476058>
- Supriyanto, R., & Siahaan, D. O. (2024). Sistem Informasi Anotasi Data Penelitian : Pengolahan Data. *Jurnal Teknik ITS*, 13(1), 98–103. <https://doi.org/10.12962/j23373539.v13i1.128602>
- Surbakti, A. Q., Hayami, R., & Amien, J. Al. (2021). Analisa Tanggapan Terhadap PSBB di Indonesia dengan Algoritma Decision Tree pada Twitter. *Jurnal CoSciTech (Computer Science and Information Technology)*, 2(2), 91–97. <https://doi.org/10.37859/coscitech.v2i2.2851>
- Wijaya, D. P., Murti, L. D., & Rachman, M. R. (2022). Recall dan Precision pada Online Public Access Catalog (OPAC) Dinas Arsip dan Perpustakaan Kota Bandung. *VISI PUSTAKA: Buletin Jaringan Informasi Antar Perpustakaan*, 24(1), 81–91. <https://doi.org/10.37014/visipustaka.v24i1.2915>
- Wulandari, S. (2023). Memastikan Keselamatan dan Kesehatan di Industri Konstruksi: Tantangan dan Solusi K3 yang Efektif. *ARRAZI: Scientific Journal of Health*, 1(2), 103–112. <https://journal.csspublishing.com/index.php/arrazi/article/view/255>
- Yasen, N. M., Rifka, S., Vitria, R., & Yulindon, Y. (2023). Pemanfaatan YOLO untuk Deteksi Hama dan Penyakit pada Daun Cabai Menggunakan Metode Deep Learning. *Elektron: Jurnal Ilmiah*, 15, 63–71. <https://doi.org/10.30630/eji.0.0.397>

