

Deteksi Penggunaan Kacamata *Safety* Berbasis YOLOv8 untuk Peningkatan Keselamatan Kerja

Wildan Eky Santoso¹, Arnisa Stefanie², Reni Rahmadewi³

^{1,2,3}Program Studi Teknik Elektro, Fakultas Teknik, Universitas Singaperbangsa Karawang, Indonesia

¹ekywildan546@gmail.com, ²arnisa.stefanie@staff.unsika.ac.id, ³reni.rahmadewi@ft.unsika.ac.id

Abstrak

Penggunaan Alat Pelindung Diri (APD) seperti kacamata *safety* merupakan aspek penting dalam menjaga keselamatan kerja di lingkungan industri karena melindungi mata dari debu, serpihan material, dan zat berbahaya. Namun, tingkat kepatuhan pekerja terhadap penggunaan APD masih rendah dan pengawasan manual dinilai kurang efektif karena bergantung pada kemampuan observasi manusia serta berpotensi menimbulkan kelalaian. Penelitian ini bertujuan mengembangkan sistem deteksi penggunaan kacamata *safety* secara *real-time* menggunakan metode YOLOv8 berbasis *deep learning*. Dataset dikumpulkan secara mandiri sebanyak 857 citra yang dibagi ke dalam dua kelas, yaitu *Safety* dan *No_Safety*, dengan variasi sudut pengambilan gambar, jarak objek terhadap kamera, dan latar belakang. Pelabelan data dilakukan menggunakan platform Roboflow dengan metode *bounding box* sebanyak 1074 anotasi. Model dilatih pada Google Colab menggunakan parameter 100 epoch, ukuran citra 640×640 piksel, dan *learning rate* yang disesuaikan. Implementasi sistem dilakukan menggunakan Python dan Visual Studio Code melalui input *webcam* untuk mendeteksi objek secara langsung. Evaluasi kinerja model menggunakan *confusion matrix* dengan metrik akurasi, presisi, *recall*, dan *F1-score*. Hasil penelitian menunjukkan bahwa model YOLOv8 mampu mendeteksi penggunaan kacamata *safety* dengan akurasi 78%, presisi 77%, *recall* 86%, dan *F1-score* 81%. Sistem juga mampu mendeteksi beberapa wajah dalam satu *frame* secara bersamaan. Secara keseluruhan, model menunjukkan performa cukup baik, namun masih diperlukan penambahan variasi *dataset* dan augmentasi data guna meningkatkan akurasi serta mengurangi kesalahan klasifikasi.

Kata kunci: keselamatan kerja, alat pelindung diri (APD), deteksi kacamata *safety*, YOLOv8, *deep learning*

1. Pendahuluan

Keselamatan kerja merupakan aspek penting dalam lingkungan industri karena berperan dalam menciptakan kondisi kerja yang aman, tertib, dan produktif (Ladys, 2025). Penerapan sistem Keselamatan dan Kesehatan Kerja (K3) yang baik juga menjadi indikator penting dalam menilai tingkat kedisiplinan dan profesionalisme pekerja dalam menjalankan tugasnya (Ayu Marsela, 2024). Salah satu bentuk perlindungan yang wajib diterapkan adalah penggunaan Alat Pelindung Diri (APD), seperti kacamata *safety* yang berfungsi melindungi mata dari debu, serpihan material, maupun paparan zat berbahaya (Yani, 2024). Namun, pada praktiknya masih banyak ditemukan pekerja yang mengabaikan penggunaan APD dengan alasan ketidaknyamanan atau rendahnya kesadaran terhadap pentingnya keselamatan kerja (Sarbiah, 2023). Kondisi ini dapat meningkatkan risiko terjadinya kecelakaan kerja, terutama pada area dengan tingkat bahaya tinggi (Hikmi, 2022). Selain itu, pengawasan penggunaan APD yang masih dilakukan secara manual dinilai kurang efektif karena bergantung pada kemampuan observasi manusia yang terbatas serta berpotensi menimbulkan kesalahan atau

kelalaian dalam pengawasan (Ihsan, 2025).

Perkembangan teknologi saat ini memungkinkan adanya solusi otomatis melalui pemanfaatan *computer vision* berbasis *deep learning* untuk mendeteksi penggunaan APD (Mandela, 2025). Beberapa metode deteksi objek seperti R-CNN, SSD, dan YOLO telah banyak digunakan untuk mengenali objek pada gambar maupun video (Achmad, 2025). Di antara metode tersebut, YOLO memiliki keunggulan dalam hal kecepatan karena menggunakan pendekatan *single-stage detector*, sehingga cocok digunakan untuk sistem *real-time* (Dzaky, 2023).

Perkembangan metode YOLO terus mengalami peningkatan hingga versi terbaru, yaitu YOLOv8 (Terven, 2023). YOLOv8 memiliki performa deteksi yang lebih baik, terutama untuk objek berukuran kecil, dengan tetap mempertahankan kecepatan dan efisiensi komputasi (Azzahra, 2024).

Beberapa penelitian terdahulu telah menerapkan metode YOLO untuk mendeteksi penggunaan APD pada lingkungan kerja. Penelitian yang dilakukan oleh (RISHAD, 2025) menggunakan metode YOLO untuk mendeteksi helm dan rompi keselamatan secara *real-time* dengan tingkat akurasi yang cukup

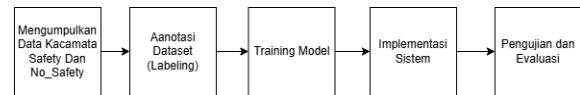
baik. Penelitian lain oleh (Calvin, 2024) memanfaatkan YOLOv7 untuk pengawasan kepatuhan penggunaan APD pada area industri dan menunjukkan peningkatan performa deteksi dibandingkan versi sebelumnya. Selain itu, penelitian oleh (Azzahra, 2024) menggunakan YOLOv8 pada sistem deteksi APD dan memperoleh hasil yang lebih baik terutama pada proses inferensi *real-time*. Meskipun demikian, sebagian besar penelitian sebelumnya masih berfokus pada deteksi APD secara umum, seperti helm dan rompi, sedangkan deteksi khusus pada kacamata *safety* masih jarang dilakukan (RISHAD, 2025). Padahal, kacamata *safety* memiliki tingkat kesulitan tersendiri dalam proses deteksi karena ukurannya yang kecil dan sering tertutup oleh posisi wajah atau kondisi pencahayaan (Djaohar, 2022).

Penelitian ini difokuskan pada deteksi penggunaan kacamata *safety* sebagai salah satu APD yang penting digunakan oleh pekerja di lingkungan industri (Syahrizal, 2021). Berbeda dengan penelitian sebelumnya yang lebih banyak mendeteksi APD secara umum, penelitian ini lebih menekankan pada deteksi kacamata *safety* karena objek tersebut memiliki ukuran yang kecil sehingga lebih sulit dikenali oleh sistem deteksi. Selain itu, penggunaan kacamata *safety* juga sering terhalang oleh posisi wajah maupun kondisi pencahayaan yang berubah-ubah sehingga dibutuhkan metode deteksi yang lebih optimal (Wahiddin, 2025).

Dalam penelitian ini digunakan *dataset* yang disesuaikan dengan kondisi lingkungan industri, seperti variasi pencahayaan, posisi pekerja, sudut pengambilan gambar, dan kondisi penggunaan kacamata *safety* di area kerja (Reza, 2025). Penggunaan *dataset* tersebut bertujuan agar model yang dikembangkan dapat mengenali objek dengan lebih baik pada kondisi nyata di lapangan. Selain itu, penelitian ini juga melakukan optimasi pada model YOLOv8 untuk meningkatkan kemampuan deteksi objek kecil, khususnya kacamata *safety* (Rahma, 2024). Dengan optimasi tersebut diharapkan model mampu menghasilkan akurasi deteksi yang lebih baik dan lebih stabil dibandingkan penelitian sebelumnya.

Sistem yang dikembangkan juga dirancang agar mampu bekerja secara *real-time* dan mendeteksi lebih dari satu objek dalam satu *frame* atau multi-object detection (Dennis, 2025). Kemampuan ini penting karena lingkungan industri memiliki banyak pekerja yang beraktivitas secara bersamaan, sehingga sistem harus mampu mendeteksi penggunaan kacamata *safety* secara cepat dan otomatis untuk meningkatkan kepatuhan keselamatan kerja.

Oleh karena itu penelitian ini mengusulkan pengembangan sistem deteksi penggunaan kacamata *safety* secara khusus dengan menggunakan metode YOLOv8 yang mampu bekerja secara *real-time*. Penelitian ini tidak hanya berfokus pada penerapan *model* saja, tetapi juga mencoba meningkatkan kemampuan deteksi terutama pada objek berukuran ke-



Gambar 1. Diagram Alur Penelitian

cil dengan menggunakan *dataset* yang disesuaikan dengan kondisi lingkungan kerja. Dengan adanya sistem ini, diharapkan hasil deteksi yang diperoleh menjadi lebih akurat dan stabil dibandingkan dengan penelitian sebelumnya. Permasalahan yang diangkat dalam penelitian ini adalah bagaimana merancang sistem yang dapat mendeteksi penggunaan kacamata *safety* secara otomatis dengan tingkat akurasi yang tinggi dalam berbagai kondisi. Oleh karena itu, tujuan dari penelitian ini adalah mengembangkan serta menguji kinerja *model* YOLOv8 dalam mendeteksi kacamata *safety* agar dapat membantu meningkatkan kepatuhan pekerja terhadap aturan keselamatan kerja.

2. Metode

Penelitian ini dilaksanakan melalui beberapa tahapan yang tersusun secara sistematis dengan tujuan untuk mengembangkan sistem deteksi penggunaan kacamata *safety* berbasis *deep learning*, dengan penelitian alur kerja pada Gambar 1.

Diagram alur pada Gambar 1 Proses penelitian dilakukan secara bertahap, dimulai dari pengumpulan data hingga tahap evaluasi sistem. Adapun tahapan yang dilakukan meliputi pengumpulan *dataset*, proses pelabelan data, pelatihan *model*, implementasi sistem, serta pengujian dan evaluasi kinerja system.

2.1 Dataset

Tahap awal dalam penelitian ini adalah proses pengumpulan *dataset* yang akan digunakan sebagai data latih 80%, data *valid* 10%, dan data uji 10% untuk mendukung proses pelatihan serta evaluasi model secara optimal. *Dataset* dikumpulkan secara mandiri dengan memanfaatkan berbagai sumber, baik dari pengambilan gambar secara langsung menggunakan kamera maupun dari *dataset* pendukung lainnya. Data yang diperoleh kemudian dikelompokkan ke dalam dua kelas utama, yaitu pengguna yang menggunakan kacamata *safety* dan yang tidak menggunakan kacamata *safety*. Dalam proses pengambilan data, peneliti juga mempertimbangkan variasi kondisi lingkungan agar *dataset* yang dihasilkan lebih beragam. Variasi tersebut meliputi perbedaan sudut pengambilan gambar, jarak objek terhadap kamera, serta latar belakang. Hal ini dilakukan dengan tujuan agar *model* yang dikembangkan memiliki kemampuan generalisasi yang baik sehingga dapat digunakan secara optimal pada kondisi nyata di lapangan.

2.2 Pelabelan Dataset

Setelah proses pengumpulan *dataset* selesai dilakukan, tahap berikutnya adalah pelabelan data (*annotation*). Tahapan ini bertujuan untuk memberikan informasi mengenai lokasi objek pada setiap gambar dengan menggunakan metode *bounding box*. Proses pelabelan dilakukan dengan bantuan *platform* Roboflow. Pada tahap ini, setiap objek *kacamata safety* yang terdapat dalam gambar diberi *label* sesuai dengan kelas yang telah ditentukan. Selain itu, *dataset* yang telah dilabeli juga dibagi menjadi tiga bagian utama, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*), guna mendukung proses pelatihan dan evaluasi *model* secara optimal.

2.3 Training Model

Tahap pelatihan *model* dilakukan menggunakan algoritma YOLOv8 pada *platform* Google Colab dengan memanfaatkan *GPU* untuk mempercepat proses komputasi. *Model* dilatih menggunakan *dataset* berformat YAML dengan parameter utama seperti 100 *epoch*, ukuran citra 640×640 piksel, serta *learning rate* yang disesuaikan. Hasil pelatihan ditampilkan dalam bentuk tabel *result* dan grafik perkembangan setiap *epoch* yang mencakup fase pelatihan dan validasi. Metrik yang diamati meliputi *box_loss* untuk kesalahan pendeteksian *bounding box* dan *cls_loss* untuk kesalahan klasifikasi objek, di mana penurunan nilai *loss* menunjukkan peningkatan performa *model* dalam mendeteksi *kacamata safety*.

2.4 Implementasi Sistem

Model yang telah dilatih kemudian diimplementasikan ke dalam sistem deteksi berbasis *Python* menggunakan *Visual Studio Code*, yang dirancang untuk melakukan deteksi secara *real-time* melalui input kamera *webcam*. Proses implementasi dimulai dengan memuat *model* hasil pelatihan, dilanjutkan dengan pengambilan data video secara langsung, kemudian sistem memproses setiap *frame* untuk mendeteksi objek menggunakan algoritma YOLOv8. Hasil deteksi ditampilkan dalam bentuk *bounding box* dan *label* pada objek yang terdeteksi, sehingga sistem dapat secara otomatis memberikan informasi apakah seseorang menggunakan atau tidak menggunakan *kacamata safety* secara langsung.

2.5 Pengujian dan Evaluasi

Pengujian dan evaluasi dilakukan untuk menilai kinerja *model* menggunakan data pengujian. Parameter yang digunakan meliputi *accuracy* (akurasi), *precision*, *recall*, dan sebagai metrik utama deteksi objek. Selain itu, evaluasi juga dilakukan menggunakan *confusion matrix* yang terdiri dari nilai *true positive*

(TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN), sehingga dapat memberikan gambaran performa *model* secara lebih detail dan menyeluruh (Pristian, 2022).

Akurasi merupakan metrik yang digunakan untuk menunjukkan persentase keseluruhan data yang berhasil diklasifikasikan dengan benar, baik pada kelas positif maupun negatif. Perhitungan akurasi dilakukan dengan membandingkan jumlah seluruh prediksi yang benar terhadap total data yang digunakan dalam proses pengujian. Persamaan umum untuk menghitung akurasi yaitu (1):

$$\text{Akurasi} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (1)$$

Seluruh nilai dalam persamaan tersebut dapat diperoleh dari *confusion matrix*, karena matriks ini menampilkan komponen utama seperti TP, TN, FP, dan FN yang digunakan untuk menghitung setiap bagian metrik evaluasi (Azzahra, 2024).

Presisi (*Precision*) digunakan untuk mengukur ketepatan *model* dalam memprediksi kelas positif. Nilai ini menunjukkan perbandingan antara prediksi positif yang benar dengan seluruh prediksi positif. Dalam deteksi objek, presisi menggambarkan jumlah *bounding box* yang benar-benar sesuai dengan objek yang dideteksi. Persamaan umum untuk menghitung presisi yaitu (2):

$$\text{Presisi} = \frac{TP}{(TP + FP)} \quad (2)$$

True Positive merupakan jumlah objek yang berhasil diprediksi dengan benar sebagai kelas positif oleh *model*, sedangkan *False Positive* adalah jumlah objek yang diprediksi sebagai positif, tetapi pada kenyataannya bukan objek tersebut (Azzahra, 2024).

Recall atau *sensitivity* merupakan metrik yang menunjukkan kemampuan *model* dalam mendeteksi seluruh data yang benar-benar berlabel positif. Persamaan umum untuk menghitung *recall* yaitu (3):

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (3)$$

Nilai ini menggambarkan seberapa banyak data aktual positif yang berhasil diprediksi dengan benar oleh *model*, sehingga mencerminkan tingkat keberhasilan *model* dalam menemukan kembali objek atau informasi yang relevan (Pristian, 2022).

F1-score merupakan salah satu metrik evaluasi yang digunakan untuk mengukur keseimbangan antara nilai *precision* dan *recall* dengan menggunakan rata-rata harmonik. Persamaan umum untuk menghitung *recall* yaitu (4):

$$F1 \text{ Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (4)$$

Dalam perhitungannya, nilai *true negative* tidak diperhitungkan. Metrik ini digunakan untuk menggambarkan seberapa baik dan stabil kinerja *model* dalam melakukan proses klasifikasi (Prasetyo, 2023).

Hasil evaluasi digunakan sebagai acuan dalam

Gambar 2. Hasil Pelabelan *Dataset*

menilai kinerja *model*. Selain itu, hasil tersebut juga menjadi dasar dalam proses analisis. Selanjutnya, digunakan untuk pengembangan dan perbaikan *model* ke depannya.

3. Hasil dan Pembahasan

3.1 Hasil Penelitian *Dataset*

Pada tahap pengumpulan *dataset* dalam penelitian ini, diperoleh sebanyak 857 citra sebelum proses pelabelan (*annotation*) yang selanjutnya digunakan sebagai data latih dan data uji. *Dataset* tersebut terdiri atas 437 citra pengguna yang menggunakan kacamata *safety* dan 420 citra pengguna yang tidak menggunakan kacamata *safety*. Komposisi data yang relatif seimbang ini bertujuan untuk meminimalkan bias model terhadap salah satu kelas sehingga hasil deteksi dapat menjadi lebih optimal. Selain itu, *dataset* yang digunakan memiliki variasi kondisi, seperti perbedaan sudut pengambilan gambar, jarak objek terhadap kamera, serta latar belakang yang beragam. Variasi tersebut penting untuk meningkatkan kemampuan generalisasi model agar sistem deteksi mampu bekerja dengan baik pada berbagai kondisi nyata di lingkungan kerja.

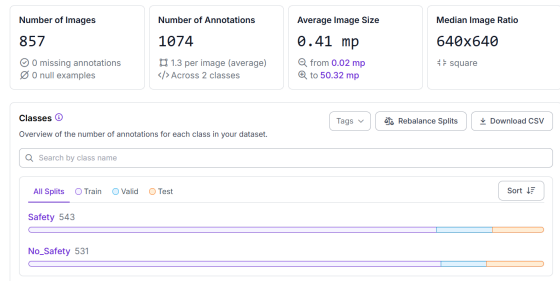
3.2 Hasil Pelabelan *Dataset*

Pada tahap pelabelan data, seluruh *dataset* yang telah dikumpulkan kemudian dilakukan proses anotasi menggunakan platform Roboflow dengan metode *bounding box* untuk menandai objek pada setiap citra terlihat pada Gambar 2 berikut.

Berdasarkan hasil pelabelan pada Gambar 2, terlihat bahwa setiap objek wajah dalam satu citra dapat memiliki *label* yang berbeda sesuai dengan kondisi penggunaan kacamata *safety*, yaitu *Safety* yang ditandai dengan kotak berwarna hijau dan *No_Safety* yang ditandai dengan kotak berwarna merah. Proses pelabelan dilakukan secara detail pada setiap objek, sehingga dalam satu citra dapat terdapat lebih dari satu objek dengan *label* yang berbeda. Selain itu,

Tabel 1. Komposisi *Dataset* untuk Pelatihan dan Evaluasi *Model*

Class	Train	Valid	Test
<i>Safety</i>	430	59	54
<i>No Safety</i>	425	47	59
Total	1.074		

Gambar 3. *Dataset Analytics*

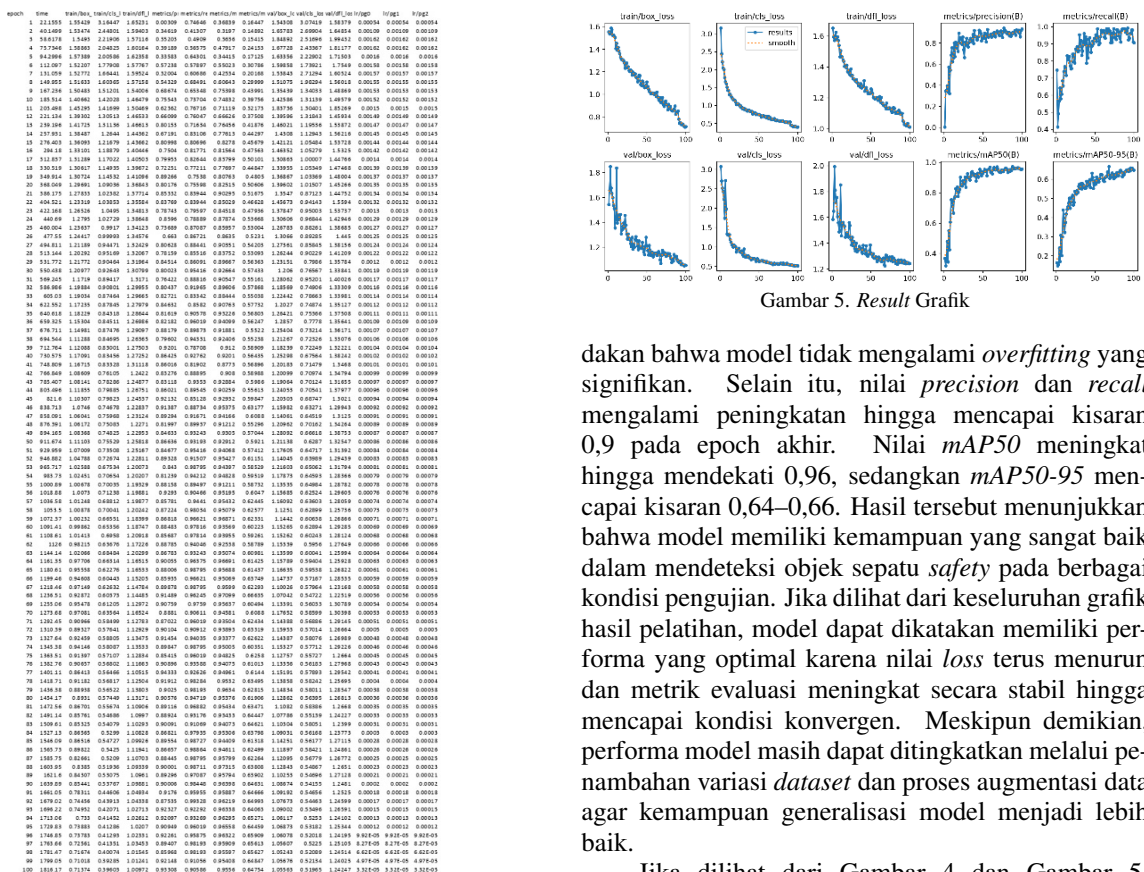
dataset yang telah dilabeli kemudian dibagi menjadi tiga bagian, yaitu data latih (*training*), data validasi (*validation*), dan data uji (*testing*) untuk mendukung proses pelatihan dan evaluasi *model*. Adapun distribusi *dataset* dapat dilihat pada Tabel 1 dan Gambar 3.

Berdasarkan Tabel 1 dan Gambar 3, *dataset* yang digunakan dalam penelitian ini terdiri atas 857 citra dengan total 1.074 anotasi yang terbagi ke dalam dua kelas, yaitu *Safety* sebanyak 543 anotasi dan *No Safety* sebanyak 531 anotasi. Komposisi *dataset* yang relatif seimbang ini bertujuan untuk mengurangi bias model terhadap salah satu kelas selama proses pelatihan. Pembagian *dataset* dilakukan ke dalam data latih, validasi, dan pengujian, di mana kelas *Safety* terdiri atas 430 data *train*, 59 data validasi, dan 54 data data test, sedangkan kelas *No Safety* terdiri atas 425 data *train*, 47 data validasi, dan 59 data test. Selain itu, *dataset* memiliki rata-rata ukuran citra sebesar 0,41 megapiksel dengan resolusi median 640×640 piksel berbentuk persegi (*square*), sehingga sesuai untuk proses pelatihan model YOLO. Tidak adanya *missing annotation* maupun *null examples* juga menunjukkan bahwa kualitas *dataset* telah terjaga dengan baik dan layak digunakan untuk proses pelatihan serta evaluasi model deteksi penggunaan kacamata *safety*.

3.3 Hasil *Training Model*

Pada tahap pelatihan *model*, dilakukan proses *training* menggunakan algoritma YOLOv8 selama 100 *epoch* untuk mengetahui perkembangan kinerja *model* dalam mendeteksi kacamata *safety*, hasil pelatihan selama 100 *epoch* ditampilkan pada Gambar 4.

Berdasarkan hasil pelatihan model YOLOv8 selama 100 *epoch*, terlihat pada Gambar 4 bahwa nilai *train loss* dan *validation loss* mengalami penurunan secara bertahap dari *epoch* awal hingga akhir, yang menunjukkan bahwa model semakin baik dalam



Gambar 5. Result Grafik

dakan bahwa model tidak mengalami *overfitting* yang signifikan. Selain itu, nilai *precision* dan *recall* mengalami peningkatan hingga mencapai kisaran 0,9 pada epoch akhir. Nilai *mAP50* meningkat hingga mendekati 0,96, sedangkan *mAP50-95* mencapai kisaran 0,64–0,66. Hasil tersebut menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi objek sepatu *safety* pada berbagai kondisi pengujian. Jika dilihat dari keseluruhan grafik hasil pelatihan, model dapat dikatakan memiliki performa yang optimal karena nilai *loss* terus menurun dan metrik evaluasi meningkat secara stabil hingga mencapai kondisi konvergen. Meskipun demikian, performa model masih dapat ditingkatkan melalui penambahan variasi *dataset* dan proses augmentasi data agar kemampuan generalisasi model menjadi lebih baik.

Jika dilihat dari Gambar 4 dan Gambar 5, hasil pelatihan model menunjukkan bahwa model YOLOv8 sudah memiliki performa yang baik dalam mendeteksi objek sepatu *safety*. Hal ini dapat dilihat dari nilai *loss* yang terus menurun pada setiap epoch, serta nilai *precision*, *recall*, dan *mAP* yang mengalami peningkatan dan cenderung stabil pada akhir pelatihan. Selain itu, model juga tidak mengalami *overfitting* yang signifikan karena nilai *validation loss* ikut menurun dan stabil. Meskipun hasil yang diperoleh sudah cukup baik, performa model masih dapat ditingkatkan lagi dengan menambahkan jumlah *dataset*, memperbanyak variasi data, serta melakukan augmentasi agar model dapat mengenali objek dengan lebih optimal pada berbagai kondisi.

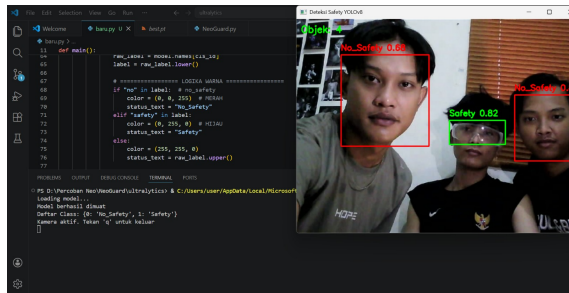
Hasil penelitian ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa model YOLOv8 memiliki performa yang baik dalam deteksi objek secara umum, namun masih mengalami keterbatasan dalam mendeteksi objek dengan ukuran kecil, yang berdampak pada nilai *recall* dan *mAP* yang tidak maksimal. Jika dibandingkan dengan penelitian sebelumnya yang menggunakan pelatihan selama 170 epoch dan memperoleh nilai *precision* sebesar 0,877, *recall* sebesar 0,768, serta *mAP50* sebesar 0,852, maka hasil pada penelitian ini masih berada pada kisaran yang sebanding meskipun menggunakan jumlah epoch yang lebih sedikit. Hal ini menunjukkan bahwa model yang dikembangkan telah mampu mencapai performa yang cukup baik. Namun demikian, berdasarkan hasil *confusion matrix*, masih terdapat

mempelajari karakteristik objek dan mengurangi kesalahan prediksi. Di sisi lain, nilai *precision*, *recall*, *mAP50*, dan *mAP50-95* mengalami peningkatan yang cukup signifikan dan cenderung stabil pada epoch akhir. Pada epoch ke-100, model memperoleh nilai *precision* sebesar 0,93308, *recall* sebesar 0,90586, *mAP50* sebesar 0,95560, dan *mAP50-95* sebesar 0,64754, sehingga menunjukkan kemampuan deteksi objek yang sangat baik. Stabilitasnya nilai evaluasi pada epoch akhir menandakan bahwa model telah mencapai kondisi konvergen, yaitu kondisi ketika peningkatan performa model sudah tidak terlalu signifikan meskipun jumlah epoch ditambah, sehingga model dinilai optimal dan siap digunakan untuk proses deteksi secara *real-time*.

Selanjutnya pada grafik metrik evaluasi yang ditunjukkan pada Gambar 5, terlihat bahwa nilai *train loss* yang terdiri dari *box_loss*, *cls_loss*, dan *dfl_loss* mengalami penurunan secara bertahap dari awal hingga akhir epoch. Penurunan tersebut menunjukkan bahwa model semakin mampu mempelajari pola pada *dataset*, baik dalam menentukan lokasi objek maupun dalam mengklasifikasikan objek dengan lebih akurat.

Pada Gambar 5 tersebut bagian *validation loss* juga terlihat adanya fluktuasi pada beberapa epoch awal, namun secara keseluruhan nilai *val/box_loss*, *val/cls_loss*, dan *val/dfl_loss* terus menurun dan cenderung stabil pada epoch akhir, sehingga menan-

Gambar 4. Result 100 Epoch



Gambar 6. Hasil Implementasi Sistem

kesalahan deteksi berupa *false negative* (FN), yang menunjukkan bahwa model belum sepenuhnya sensitif dalam mendeteksi objek kacamata pada kondisi tertentu. Oleh karena itu, diperlukan peningkatan lebih lanjut seperti penambahan variasi *dataset*, penerapan teknik augmentasi data, serta penyesuaian parameter pelatihan agar performa model dapat menjadi lebih optimal (Swandaru, 2025).

3.4 Hasil Implementasi Sistem

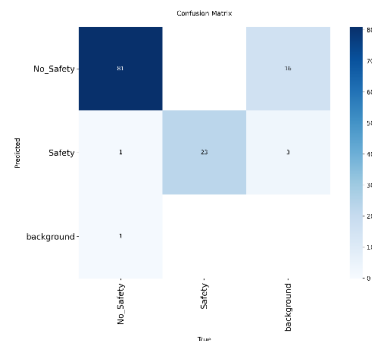
Pada tahap pengujian dan evaluasi, sistem diuji menggunakan kamera untuk melihat kinerja *model* dalam mendeteksi objek. Berdasarkan hasil pada Gambar 6, sistem mampu mendeteksi beberapa wajah dalam satu *frame* dan mengklasifikasikannya ke dalam dua kelas, yaitu *Safety* dan *No_Safety*. Terlihat bahwa dua objek terdeteksi sebagai *No_Safety* dengan nilai *confidence* sebesar 0.68 dan 0.81, sedangkan satu objek lainnya terdeteksi sebagai *Safety* dengan nilai *confidence* sebesar 0.82. Hal ini menunjukkan bahwa *model* sudah mampu membedakan penggunaan kacamata *safety* dengan cukup baik.

Jika dilihat dari hasil deteksinya terlihat pada Gambar 6, *model* memiliki kemampuan yang cukup baik dalam mengenali objek sehingga nilai *precision* dan *recall* dapat dikatakan sudah cukup optimal. Meskipun demikian, perbedaan nilai *confidence* pada setiap objek menunjukkan bahwa performa *model* masih dapat ditingkatkan, terutama pada kondisi tertentu seperti pencahayaan dan posisi wajah. Secara keseluruhan, hasil pengujian ini menunjukkan bahwa sistem sudah dapat berjalan dengan baik dan mampu melakukan deteksi penggunaan kacamata *safety*.

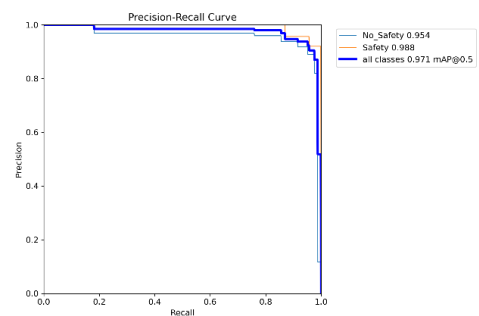
3.5 Hasil Pengujian dan Evaluasi

Model sistem yang telah melalui tahap pelatihan akan menghasilkan *confusion matrix* sebagai bentuk evaluasi kinerja *model*. Selain itu, *model* juga menampilkan visualisasi berupa *Precision-Recall Curve* dan *Confidence Curve* yang digunakan untuk melihat tingkat akurasi dan keseimbangan performa *model* pada berbagai tingkat kepercayaan terlihat pada Gambar 7.

Confusion Matrix pada Gambar 7 merupakan matriks berukuran 3×3 yang digunakan untuk menunjukkan hubungan antara kelas sebenarnya pada



Gambar 7. Confusion Matrix



Gambar 8. Kurva Grafik Precision-Recall Curve

objek dengan kelas hasil prediksi yang dihasilkan oleh *model*. Melalui *confusion matrix* ini, dapat diketahui kinerja *model* dalam melakukan klasifikasi dengan melihat beberapa komponen utama, yaitu *True Positive*, *False Positive*, dan *False Negative*, yang menggambarkan tingkat keberhasilan serta kesalahan prediksi yang dilakukan oleh *model*.

True Positif (TP) = 81

False Negatif (FN) = 16

False Positif (FP) = 1 + 1 = 2

True Negatif (TN) = 23 + 3 = 26

Akurasi prediksi *model* menggunakan persamaan (1):

$$\text{Akurasi} = \frac{(81 + 26)}{(81 + 2 + 16 + 26)} = 0,85 \text{ atau } 85\%$$

Presisi prediksi dapat dihitung menggunakan persamaan (2):

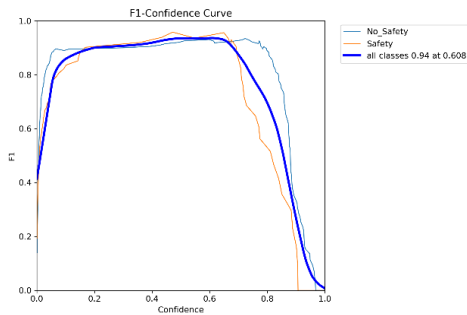
$$\text{Presisi} = \frac{81}{(81 + 2)} = 0,97 \text{ atau } 97\%$$

Recall dapat dihitung menggunakan persamaan (3):

$$\text{Recall} = \frac{81}{(81 + 16)} = 0,83 \text{ atau } 83\%$$

$$F1 \text{ Score} = 2 \times \frac{0,97 \times 0,83}{0,97 + 0,83} = 0,89 \text{ atau } 89\%$$

Confusion Matrix pada Gambar 7 menunjukkan hubungan antara kelas sebenarnya dengan hasil prediksi *model*. Berdasarkan hasil pengujian diperoleh nilai *True Positive* (TP) sebesar 81, *False Negative* (FN) sebesar 16, *False Positive* (FP) sebesar 2, dan *True Negative* (TN) sebesar 26. Dari hasil tersebut diperoleh nilai akurasi sebesar 85%, presisi sebesar 97%, dan *recall* sebesar 83%. Nilai presisi yang



Gambar 9. Kurva Grafik F1-Confidence Curve

tinggi menunjukkan bahwa model mampu melakukan prediksi dengan cukup tepat dan hanya menghasilkan sedikit kesalahan deteksi (*false positive*). Namun, nilai *recall* yang lebih rendah dibandingkan presisi menunjukkan bahwa masih terdapat beberapa objek yang tidak berhasil terdeteksi oleh model (*false negative*).

Kesalahan deteksi tersebut dapat dipengaruhi oleh beberapa faktor, seperti kondisi pencahayaan yang kurang stabil sehingga objek terlihat kurang jelas pada kamera. Selain itu, ukuran objek yang kecil juga menyebabkan detail objek sulit dikenali oleh model. Faktor sudut pengambilan gambar atau posisi wajah yang berbeda-beda juga memengaruhi hasil deteksi karena model kesulitan mengenali karakteristik objek pada sudut tertentu. Hal inilah yang menyebabkan masih terdapat beberapa objek yang gagal terdeteksi sehingga nilai *recall* menjadi lebih rendah dibandingkan nilai presisi.

Berdasarkan grafik *Precision-Recall Curve* terlihat pada Gambar 8, dimana performa model terlihat cukup baik karena nilai presisi dan *recall* cenderung stabil pada sebagian besar pengujian. Hal ini menunjukkan bahwa model mampu mendeteksi objek dengan cukup konsisten meskipun masih terdapat beberapa kesalahan klasifikasi.

Pada Gambar 9 grafik *F1-Confidence Curve*, terlihat bahwa nilai *F1-score* meningkat seiring bertambahnya nilai *confidence* dan mencapai titik optimal pada *confidence* sekitar 0,6 dengan nilai *F1-score* sebesar 0,94. Hal ini menunjukkan bahwa model memiliki keseimbangan yang baik antara nilai presisi dan *recall* pada titik tersebut. Namun, setelah melewati titik optimal, nilai *F1-score* mulai menurun karena model menjadi lebih selektif dalam mendeteksi objek. Secara keseluruhan, model sudah memiliki performa yang cukup baik dalam mendeteksi objek sepatu *safety*, namun masih perlu dilakukan pengembangan lebih lanjut, seperti penambahan jumlah *dataset* dan variasi kondisi pengujian agar hasil deteksi menjadi lebih optimal.

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi penggunaan kacamata *safety* berbasis

YOLOv8 berhasil dikembangkan dan memiliki performa yang cukup baik dalam mendeteksi penggunaan kacamata *safety*. Model dilatih menggunakan 857 citra dengan total 1.074 anotasi yang terbagi ke dalam dua kelas, yaitu *Safety* dan *No Safety*. Hasil pelatihan selama 100 epoch menunjukkan bahwa nilai *train loss* dan *validation loss* mengalami penurunan secara bertahap, sedangkan nilai *precision*, *recall*, *mAP50*, dan *mAP50-95* mengalami peningkatan dan cenderung stabil pada epoch akhir, sehingga menandakan bahwa model telah mencapai kondisi konvergen tanpa mengalami *overfitting* yang signifikan. Pada hasil evaluasi diperoleh nilai akurasi sebesar 85%, presisi 97%, *recall* 83%, dan *F1-score* 89%, yang menunjukkan bahwa model mampu melakukan deteksi objek dengan cukup akurat. Nilai presisi yang lebih tinggi dibandingkan *recall* menunjukkan bahwa model sangat baik dalam melakukan prediksi objek secara tepat, meskipun masih terdapat beberapa objek yang gagal terdeteksi akibat faktor pencahayaan, ukuran objek yang kecil, serta sudut pengambilan gambar yang berbeda-beda. Selain itu, implementasi sistem menggunakan kamera menunjukkan bahwa model mampu mendeteksi beberapa wajah dalam satu *frame* secara bersamaan dengan nilai *confidence* yang bervariasi. Hasil penelitian ini juga menunjukkan bahwa algoritma YOLOv8 memiliki kemampuan yang baik dalam proses deteksi objek secara *real-time*. Dengan demikian, sistem yang dikembangkan dinilai telah mampu bekerja dengan baik dalam mendeteksi penggunaan kacamata *safety* dan berpotensi untuk diterapkan sebagai pendukung keselamatan kerja di lingkungan industri.

Berdasarkan hasil tersebut, penelitian selanjutnya disarankan untuk melakukan pengembangan lebih lanjut guna meningkatkan performa sistem. Beberapa upaya yang dapat dilakukan antara lain menambah jumlah dan variasi *dataset*, khususnya pada kondisi pencahayaan yang beragam dan sudut pengambilan gambar yang lebih kompleks, serta melakukan augmentasi data untuk meningkatkan kemampuan generalisasi *model*. Selain itu, optimasi parameter pelatihan dan eksplorasi metode atau arsitektur lain yang lebih optimal juga dapat menjadi alternatif untuk meningkatkan akurasi deteksi. Pengembangan sistem ke tahap implementasi yang lebih luas, seperti integrasi dengan sistem *monitoring* berbasis IoT atau CCTV industri, juga menjadi peluang penelitian lanjutan agar sistem dapat diterapkan secara lebih efektif dalam mendukung pengawasan penggunaan APD di lingkungan kerja.

Daftar Pustaka

Achmad, M. H. (2025): *Analisis kinerja model deteksi objek YOLO, SSD, dan Faster R-CNN pada citra penglihatan malam untuk pengenalan tindak kejahatan*, Jurnal Teknologi Informasi dan

- Ilmu Komputer, 12(1), 145–152. <https://doi.org/10.25126/jtiik.2025128409>
- Alfarizi, D. N. (2023): *Penggunaan metode YOLO pada deteksi objek: Sebuah tinjauan literatur sistematis*, 1(1), 54–63.
- Ayu Marsela. (2024): *Grafik 1 data kecelakaan kerja*, 6(1), 63–73.
- Azzahra. (2024): *Analisis kinerja deteksi gerakan dan pengenalan objek produk ritel berbasis YOLOv8*, 11(1), 43–53.
- Dennis. (2025): *Deteksi otomatis penggunaan APD pada pekerja migas menggunakan deep learning dan computer vision*, Techno.Com, 24(3), 834–842. <https://doi.org/10.62411/tc.v24i3.13449>
- Djaohar, M. (2022): *Rancang bangun pengecekan alat pelindung diri menggunakan algoritma You Only Look Once (YOLO)*.
- Hikmi, N. (2022): *Hubungan penggunaan APD dengan kejadian kecelakaan kerja pada pekerja bagian produksi PT. Kunango Jantan: The relationship between the use of personal protective equipment and work accidents in workers production at PT. Kunango Jantan*, 36, 27–32.
- Ihsan, M. A. (2025): *Hubungan pengawasan terhadap kepatuhan penggunaan*, 11(2), 400–407.
- Ladys. (2025): *Pengaruh lingkungan kerja dan keselamatan kerja terhadap kinerja karyawan bagian produksi di PT, Cahaya Murni Borneo Timur*.
- Mandela. (2025): *APD multiobjek pada area konstruksi berbasis computer vision dengan deep learning*.
- Prasetyo, S. Y. (2023): *[Judul artikel belum tersedia]*, Jurnal Teknologi Terpadu, 9(1), 29–34.
- Pristian. (2022): *Komparasi metode pembelajaran mesin untuk implementasi pengambilan keputusan dalam menentukan promosi jabatan karyawan*, 6(2), 622–628.
- Rahma, G. M. (2024): *Implementasi metode You Only Look Once (YOLO) v8 pada deteksi objek alat pelindung diri*, Edu Research Indonesian Institute for Corporate Learning and Studies (IICLS), 5(1), 70–80.
- Reza, T. (2025): *[Judul artikel belum lengkap: “Terhadap akurasi sistem pengenalan wajah berbasis convolutional neural network (CNN)”]*.
- Rishad. (2025): *Intisari implementasi YOLOv8n untuk deteksi alat pelindung diri: Abstract implementation of YOLOv8n for personal protective equipment detection in automatic*.
- Sarbiah, A. (2023): *Penerapan pelaksanaan keselamatan dan kesehatan kerja (K3) pada karyawan*, 15(2), 1–11.
- Swandaru, G. (2025): *Deteksi penggunaan alat pelindung diri (APD) pada pekerja berbasis web browser menggunakan YOLOv8 dan TensorFlow.js*, 1–9.
- Syahrizal. (2021): *Hubungan penggunaan alat pelindung diri (APD) dengan kesehatan mata pada pekerja pengelasan: Studi kasus di PT. X, Aceh Besar*, Jurnal Sago Gizi dan Kesehatan, 3(1), 109–113.
- Terven, J. (2023): *A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS*, 1680–1716.
- Wahiddin, D. (2025): *Penerapan algoritma convolutional neural network (CNN) untuk deteksi kaca mata safety di perusahaan manufaktur*, 9(2), 97–106.
- Yani, A. (2024): *Efektivitas pelatihan keselamatan kerja di konstruksi dan peran manajemen dalam meningkatkan kepatuhan K3: Literatur review*, 5(2), 57–66.