

Aplikasi Absensi Pengenalan Wajah dengan Menggunakan Algoritma YOLOv11

Vanness Bee¹, Ery Hartati^{2*}

^{1,2*} Program Studi Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang, Kota Palembang, Provinsi Sumatera Selatan, Indonesia.

article info

Article history:

Received 24 November 2025

Received in revised form

20 December 2025

Accepted 10 January 2026

Available online July 2026.

Keywords:

YOLOv11; Face Recognition; Attendance System.

Kata Kunci:

YOLOv11; Pengenalan Wajah; Sistem Absensi.

abstract

Manual or semi-manual attendance recording may lead to recap errors, delayed reporting, and misuse such as proxy attendance. This study develops a web-based attendance prototype leveraging You Only Look Once version 11 (YOLOv11) to perform face detection and identity recognition within a single end-to-end pipeline. The research stages include a literature review, data acquisition and pre-processing (640×640 letterbox resize and normalization), transfer-learning-based model training, and system implementation using Laravel and MySQL integrated with a Python inference service exposed via a REST API. Model performance was assessed using standard detection metrics (precision, recall, mAP@0.5, and mAP@0.5:0.95), complemented by black-box functional testing of core application modules (enrollment, attendance logging, and reporting). Internal evaluation demonstrates strong performance with precision of 0.982, recall of 0.975, mAP@0.5 of 0.987, and mAP@0.5:0.95 of 0.963. Nevertheless, performance degrades under challenging real-world conditions (extreme low-light, backlight, mask usage, and partial occlusion) and on external dataset testing, suggesting sensitivity to domain shift. Overall, the proposed system indicates practical potential for real-time attendance automation and reduced recording errors, while highlighting the need for richer, more diverse training data and cross-domain evaluation to improve generalization.

abstract

Pencatatan absensi karyawan/mahasiswa secara manual atau semi-manual berpotensi menimbulkan kesalahan rekapitulasi, keterlambatan pelaporan, serta peluang penyalahgunaan seperti titip absen. Penelitian ini mengembangkan prototipe aplikasi absensi berbasis web yang memanfaatkan You Only Look Once versi 11 (YOLOv11) untuk mendeteksi sekaligus mengenali identitas wajah dalam satu alur kerja end-to-end. Metode penelitian meliputi studi literatur, pengumpulan dan pra-pemrosesan data (letterbox resize 640×640 dan normalisasi), pelatihan model berbasis transfer learning, serta implementasi sistem menggunakan Laravel dan MySQL yang terintegrasi dengan layanan inferensi Python melalui REST API. Evaluasi dilakukan menggunakan metrik kinerja deteksi (precision, recall, mAP@0.5, mAP@0.5:0.95) serta pengujian fungsional black-box pada modul utama (enrollment, absensi, dan pelaporan). Hasil pengujian internal menunjukkan kinerja tinggi dengan precision 0,982, recall 0,975, mAP@0.5 0,987, dan mAP@0.5:0.95 0,963. Namun, performa cenderung menurun pada kondisi operasional menantang (low-light, backlight, masker, dan oklusi parsial) serta pada pengujian dataset eksternal, yang mengindikasikan adanya pengaruh domain shift. Secara keseluruhan, sistem yang diusulkan berpotensi meningkatkan efisiensi absensi real-time dan mengurangi kesalahan pencatatan, dengan kebutuhan pengayaan data dan evaluasi lintas-domain untuk meningkatkan generalisasi.

Corresponding Author. Email: ery_hartati@mdp.ac.id ^{2}.

1. Pendahuluan

Transformasi digital mendorong organisasi atau perusahaan mengadopsi sistem informasi yang lebih terintegrasi untuk mendukung proses operasional, termasuk pencatatan kehadiran. Pencatatan kehadiran yang akurat menjadi prasyarat bagi pengelolaan sumber daya manusia karena berpengaruh pada *penggajian*, perhitungan *jam kerja*, disiplin, hingga evaluasi *kinerja* (Patel *et al.*, 2025). Pada praktiknya, proses absensi masih kerap dilakukan secara manual atau semi-manual sehingga rentan terhadap kesalahan pencatatan, keterlambatan rekapitulasi, dan potensi penyalahgunaan (misalnya *titip absen*). Keterbatasan ini menjadi semakin signifikan ketika organisasi atau perusahaan membutuhkan data kehadiran yang *real-time* dan dapat diaudit. Absensi berbasis biometrik wajah dipandang sebagai alternatif yang menjanjikan karena bersifat *non kontak*, lebih nyaman bagi pengguna, dan dapat diintegrasikan dengan kamera yang tersedia (misalnya CCTV) untuk otomatisasi proses (Mahmoud *et al.*, 2024). Namun, reliabilitas sistem sangat dipengaruhi kondisi operasional seperti variasi pencahayaan, sudut pandang, jarak kamera, serta adanya halangan termasuk penggunaan masker atau wajah tertutup sebagian (Wang & Deng, 2021).

Dalam literatur *pengenalan wajah* (face recognition), pendekatan modern umumnya diposisikan pada spektrum dari metode tradisional berbasis fitur/klasifier hingga pendekatan *deep learning* berbasis CNN. Tinjauan menyeluruh menunjukkan bahwa sistem *pengenalan wajah* (face recognition) lazimnya dibangun melalui tahapan deteksi penyelarasan (alignment), ekstraksi fitur, pencocokan, dan perkembangan CNN, terutama dalam meningkatkan tahap ekstraksi fitur agar lebih tangguh terhadap variasi pose, iluminasi, ekspresi, usia, hingga oklusi (Zhang *et al.*, 2020). Pada generasi awal *deep face recognition*, pendekatan CNN tradisional sering memformulasikan pengenalan sebagai klasifikasi identitas dengan *softmax*, misalnya DeepFace yang menggabungkan penyelarasan berbasis pemodelan wajah dan CNN untuk verifikasi wajah, serta VGGFace yang memanfaatkan arsitektur CNN dalam skala data besar untuk membangun representasi fitur wajah (Parkhi *et al.*, 2015; Taigman *et al.*, 2014). Namun, formulasi klasifikasi cenderung

kurang fleksibel untuk skenario penambahan identitas baru karena sering bergantung pada kelas identitas saat pelatihan, sehingga riset berikutnya bergeser kuat ke pembelajaran penyematan/metrik (*embedding/metric learning*). Dalam *embedding*, FaceNet mempopulerkan pembelajaran representasi wajah berdimensi rendah (mis. 128-D) yang dioptimasi dengan *triplet loss*, sehingga identitas yang sama dipaksa lebih dekat daripada identitas berbeda pada ruang *embedding*, pendekatan ini memudahkan verifikasi dan identifikasi berbasis kemiripan (Schroff *et al.*, 2015). Selanjutnya, ArcFace memperkuat diskriminabilitas *embedding* melalui additive angular margin loss pada *softmax*, yang secara geometri meningkatkan separabilitas antar kelas dan kekompakan intra kelas, sehingga menjadi dasar kuat pada banyak studi lanjutan (Deng *et al.*, 2019). Meski demikian, studi tinjauan dan evaluasi pada kondisi menantang menunjukkan bahwa performa FaceNet dan ArcFace dapat menurun pada skenario dunia nyata seperti citra *pengawasan* beresolusi rendah, oklusi parsial, dan variasi pencahayaan. Studi komparatif terkini melaporkan FaceNet dan ArcFace unggul pada kondisi relatif bersih/terkendali, tetapi menghadapi pertukaran generalisasi ketika input terdegradasi (Zhang *et al.*, 2025).

Temuan ini menegaskan pentingnya pemilihan model, desain data latih, serta strategi evaluasi yang mempertimbangkan kondisi operasional nyata (misalnya penggunaan masker dan oklusi parsial) pada aplikasi absensi otomatis. Survei terkini menunjukkan bahwa peningkatan performa *pengenalan wajah* tidak hanya ditentukan oleh rancangan sistem, tetapi juga fungsi, kualitas data, strategi pelatihan/pengumpulan data, dan skenario uji yang menyerupai kondisi nyata (Yang *et al.*, 2022). Untuk tahap deteksi wajah, model *You Only Look Once* (YOLO) menjadi salah satu pilihan populer karena karakteristiknya yang cepat dan cocok untuk aplikasi nyata. Keluarga YOLO berkembang cepat. Implementasi YOLO versi 5 (YOLOv5) banyak digunakan dalam riset terapan karena kemudahan penggunaan dan efisiensi komputasi (Jocher, 2020). Perkembangan berikutnya, YOLO versi 7 (YOLOv7) memiliki desain rancangan dan strategi pelatihan berfokus untuk mendorong pertukaran akurasi kecepatan, sehingga mencapai kinerja *real-time* yang kompetitif pada berbagai skenario deteksi objek (Wang *et al.*, 2023). Selanjutnya,

YOLO versi 8 (YOLOv8) berkembang yang menawarkan optimasi dan dukungan beragam aktivitas yang lebih luas (deteksi, segmentasi, pose) dengan fokus pada efisiensi inferensi dan kemudahan penggunaan (Ultralytics, 2020). Generasi terbaru dari lini YOLO yang dikembangkan oleh Ultralytics yaitu YOLO versi 11 (YOLOv11), menargetkan pertukaran terbaik antara akurasi, kecepatan, efisiensi komputasi untuk kebutuhan *real-time* (misalnya video) (Jegham *et al.*, 2024). Dalam segi perancangan, YOLOv11 melakukan penyempurnaan pada backbone dan neck sehingga mampu meraih mean Average Precision (mAP) lebih tinggi dengan parameter lebih sedikit dibanding versi sebelumnya, serta mendukung beragam tugas (deteksi, segmentasi, pose, *Oriented Bounding Box / OBB*) dalam satu ekosistem API (*Application Programming Interface*) (Sapkota & Karkee, 2025). YOLOv11 (Ultralytics) dapat dipandang sebagai “evolusi efisiensi” dari lini YOLO modern, dibanding YOLOv5 yang populer karena implementasi PyTorch yang mudah diadopsi (Ultralytics, 2020). YOLOv11 berfokus pada penyempurnaan rancangan dan efisiensi implementasi dalam ekosistem Ultralytics. Sementara YOLOv8 memperkenalkan *anchor-free head* dan modul C2f sebagai baseline produksi yang stabil. Ultralytics menunjukkan YOLOv11 mencapai Pareto improvement atas YOLOv8 (mAP lebih tinggi dengan parameter/FLOPs lebih rendah serta percepatan CPU yang nyata), misalnya pada COCO val dengan resolusi 640, varian menengah YOLOv11 memiliki mAP50–95 lebih tinggi dengan parameter lebih kecil daripada YOLOv8 (Jegham *et al.*, 2024).

2. Metodologi Penelitian

Penelitian ini merupakan penelitian terapan yang berfokus pada pengembangan prototipe sistem absensi berbasis pengenalan wajah secara *real-time*. Sistem dibangun sebagai aplikasi web yang terintegrasi dengan model deteksi dan identifikasi wajah berbasis *deep learning*. Pemilihan pendekatan deteksi satu tahap (YOLO) selaras dengan penelitian terdahulu yang menerapkan YOLO untuk absensi berbasis wajah karena mampu memberikan inferensi cepat pada skenario kelas/organisasi (Lee & Kim, 2020; Muhammad & Mulyani, 2021; Sahputra *et al.*, 2024).

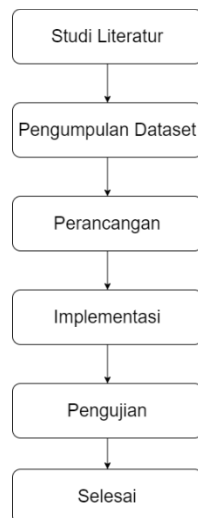
Lingkungan Pengujian Algoritma

Penelitian ini mencakup dua komponen utama, yakni (i) pelatihan model deteksi/pengenalan wajah berbasis YOLOv11 dan (ii) pengembangan aplikasi absensi berbasis web yang mengintegrasikan model tersebut. Pelatihan model dilakukan pada lingkungan komputasi berbasis Python dengan dukungan pustaka Ultralytics YOLO dan PyTorch, sedangkan aplikasi absensi dikembangkan menggunakan framework Laravel (PHP) dan basis data MySQL. Adapun perangkat keras (Hardware) yang digunakan dalam penelitian ini memiliki spesifikasi sebagai berikut: (1) Laptop / PC dengan prosesor setidaknya Intel Core i5 atau setara untuk menjamin kinerja komputasi; (2) RAM 16 GB dan SSD 512 GB, agar proses pada pelatihan model berjalan lebih stabil dan tidak mudah mengalami kendala pada memori; (3) Pelatihan model memanfaatkan GPU pada Google Colab untuk mempercepat proses pelatihan dan eksperimen model YOLOv11 sehingga waktu komputasi dapat lebih efisien; (4) Kamera Internal maupun Eksternal sebagai media input citra wajah yang akan diproses oleh sistem; dan (5) Ruang penyimpanan 256 GB atau lebih, yang digunakan untuk menampung dataset, model hasil pelatihan, serta file aplikasi. Perangkat lunak yang digunakan dalam penelitian ini meliputi (1) Sistem operasi: Windows 10/11 sebagai platform utama; (2) Bahasa pemrograman: PHP 8 untuk pengembangan aplikasi web Laravel, serta Python (≥ 3.10) untuk membangun dan menjalankan model YOLOv11; (3) Framework: Laravel sebagai framework backend, dipadukan dengan Tailwind CSS untuk mendukung perancangan antarmuka pengguna yang responsif; (4) Database: MySQL yang dimanfaatkan untuk menyimpan dan mengelola data absensi secara terstruktur; (5) Framework Deep Learning: YOLOv11 yang dijalankan dengan pustaka Ultralytics pada Python; (6) Aplikasi Pendukung: Figma (Desain UI), Visual Studio Code (Coding), Composer (manajemen dependensi Laravel) serta Anaconda / Pytorch (Lingkungan Python untuk YOLOv8). Untuk integrasi model aplikasi, layanan inferensi Python diekspos sebagai REST API sehingga dapat dipanggil dari backend Laravel secara terstandar (Jegham *et al.*, 2024).

Tahapan Penelitian

Dalam pelaksanaan penelitian, diterapkan perancangan metode untuk meminimalisir kesalahan dalam proses penelitian. Tahapan metodologi

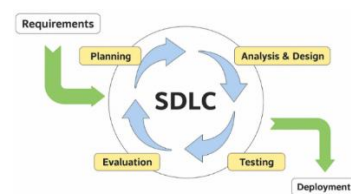
meliputi studi literatur, pengumpulan data, pra-pemrosesan dan anotasi, pelatihan model, pengembangan aplikasi (*SDLC iterative*), integrasi model dengan aplikasi web, serta pengujian dan evaluasi. Tahapan metodologi penelitian dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metodologi Penelitian

Berdasarkan Gambar 1, Tahapan metodologi penelitian yaitu: Pertama, Studi literatur: dilakukan untuk memetakan pendekatan deteksi/pengenalan wajah berbasis *deep learning* serta praktik implementasi sistem absensi digital. Fokus literatur mencakup arsitektur YOLO terkini, teknik augmentasi data, metrik evaluasi (*precision*, *recall*, *mAP*), serta strategi pencegahan *overfitting* pada pelatihan jaringan (Goodfellow *et al.*, 2016; Yang *et al.*, 2022). Kedua, Pengumpulan dataset: dengan memanfaatkan dataset publik dari Roboflow Universe (UnivBI) dengan total 1.222 citra dan 5 kelas identitas. Dataset dibagi menjadi data latih (70%), validasi (20%), dan uji (10%). Penggunaan dataset publik dari Roboflow Universe (UnivBI) dipilih untuk memastikan replikasi eksperimen, namun tetap mempertimbangkan isu bias dataset dan pergeseran domain saat diterapkan pada lingkungan berbeda (Torralla & Efros, 2011). Ketiga, Representativitas dataset terhadap kasus nyata ditinjau melalui dua mekanisme yaitu: (i) kesesuaian konteks pengambilan citra (lingkungan kampus/indoor–outdoor, variasi pencahayaan dan pose) yang mendekati skenario institusi pendidikan/perusahaan, serta (ii) rencana validasi eksternal menggunakan sampel citra dari lokasi target (pilot data) untuk mengukur generalisasi

dan/atau melakukan penyetelan bila ditemukan penurunan performa akibat domain shift (Yang *et al.*, 2022). Dengan demikian, dataset Roboflow diperlakukan sebagai baseline yang dapat ditingkatkan melalui penambahan data lokal saat implementasi. Keempat, Pra-pemrosesan mencakup penyeragaman ukuran input menggunakan *letterbox resize* ke 640×640 piksel dan normalisasi saat data loading. Anotasi mengikuti format YOLO (bounding box dan label kelas). Augmentasi diterapkan hanya pada data latih untuk meningkatkan generalisasi, sedangkan data validasi dan uji tidak diaugmentasi agar evaluasi tidak bias (Soto, 2022). Kelima, Pengembangan aplikasi dilakukan menggunakan pendekatan *Software Development Life Cycle (SDLC)* iteratif yaitu proses pengembangan dilakukan melalui siklus berulang (*iterasi*) yang mencakup perencanaan, analisis, perancangan, implementasi, pengujian, dan evaluasi sampai sistem memenuhi kebutuhan pengguna, sehingga kebutuhan operasional dapat divalidasi dan disempurnakan secara bertahap sampai tercapai fungsionalitas yang stabil (Pressman, 2020). Pendekatan iteratif dipilih karena memungkinkan penyesuaian cepat terhadap temuan selama pelatihan model dan uji fungsional aplikasi, terutama pada sistem *real-time* berbasis visi komputer. Pendekatan iteratif menekankan umpan balik pengguna dan perbaikan bertahap pada fitur sistem (Sommerville, 2016). Alur iteratif yang digunakan ditunjukkan pada Gambar 2.



Gambar 2. Model SDLC Iterative

Pelatihan Model YOLO versi 11

Model yang digunakan adalah YOLOv11 varian ringan dengan bobot awal yang sudah terlatih sebelumnya (*transfer learning*) untuk mempercepat konvergensi dan meningkatkan performa pada dataset berukuran terbatas. Pelatihan dilakukan selama 70 *epoch* dengan *batch size* 8 dan ukuran gambar 640. Optimisasi menggunakan SGD dengan *learning rate* awal $lr_0=0,01$, momentum 0,937, serta regularisasi L2 (*weight decay*=0,0005). Selain itu, diterapkan pemanasan selama 3 *epoch* dan penutupan *mosaic* pada 10 *epoch*

terakhir (*close mosaic*=10) untuk menstabilkan proses pelatihan menjelang konvergensi (Ultralytics, 2020). Konfigurasi augmentasi mengikuti parameter bawaan Ultralytics untuk tugas deteksi, meliputi perubahan HSV (*hsv_b*=0,015; *hsv_s*=0,7; *hsv_v*=0,4), translasi (*translate*=0,1), skala (*scale*=0,5), *flip horizontal* (*flip_h*=0,5), serta *mosaic* (*mosaic*=1,0). Teknik augmentasi ini umum digunakan sebagai regularisasi tingkat data untuk menambah variasi visual dan mengurangi risiko *overfitting* pada data terbatas (Ultralytics, 2020; Yang *et al.*, 2022).

Validasi Model dan Pencegahan *Overfitting*

Selain memantau kurva *loss*, penelitian ini menerapkan beberapa mekanisme pencegahan *overfitting*, yaitu: (i) regularisasi melalui *weight decay* (L2) yang menekan bobot berlebihan, (ii) augmentasi data hanya pada data latih, (iii) *early stopping* berbasis validasi dengan parameter *patience* (pelatihan dihentikan bila metrik validasi tidak membaik dalam sejumlah *epoch*), dan (iv) penutupan *mosaic* pada akhir pelatihan untuk mengurangi bunyi augmentasi saat konvergensi halus (Loshchilov & Hutter, 2017; Prechelt, 1998). Untuk meningkatkan reliabilitas estimasi performa, selain skema *hold-out* (*train/val/test*), evaluasi *robustness* dapat dilakukan melalui validasi silang stratified *k-fold* pada gabungan data latih+validasi. Teknik ini lazim digunakan untuk mengurangi variansi estimasi akurasi pada dataset yang relatif kecil (Kohavi, 1995). Pada penelitian ini, skema *hold-out* tetap digunakan sebagai pelaporan metrik utama pada data uji, sementara *k-fold* digunakan sebagai analisis tambahan untuk memastikan kestabilan performa antar pembagian data.

Implementasi Aplikasi dan Integrasi Model *Python-Laravel*

Aplikasi absensi dikembangkan berbasis web menggunakan *Laravel* dengan pola MVC (*Model View Controller*). Modul utama meliputi manajemen pengguna, pendaftaran wajah (*enrollment*), pengambilan gambar melalui kamera, pemanggilan layanan inferensi, serta pencatatan log kehadiran ke basis data *MySQL*. Integrasi model *Python* dilakukan menggunakan pendekatan *microservice*. Model *YOLOv11* dijalankan sebagai layanan *REST API* (misalnya *FastAPI*) yang menerima input citra dan mengembalikan hasil prediksi dalam format *JSON*

(*FastAPI*, 2020). Secara operasional, alur integrasi berjalan sebagai berikut: (1) pengguna membuka halaman absensi dan kamera mengambil citra/frame; (2) citra dikirim ke backend *Laravel* (*multipart/form-data* atau *base64*); (3) *Laravel* meneruskan citra ke endpoint inferensi *Python* melalui HTTP POST; (4) layanan *Python* melakukan pra-pemrosesan, inferensi *YOLOv11*, dan mengembalikan *bounding box* wajah, skor *confidence*, serta label identitas; (5) *Laravel* menerapkan ambang *confidence* (misalnya $\geq 0,25$) dan memilih prediksi terbaik; (6) identitas yang terdeteksi dipetakan ke akun pengguna pada basis data, kemudian sistem menyimpan waktu dan status kehadiran; dan (7) hasil ditampilkan kembali pada antarmuka web secara *real-time*. Pola integrasi berbasis API ini memudahkan pemeliharaan karena komponen web dan model dapat dikembangkan serta disebarluaskan secara terpisah (Ultralytics, 2020; *FastAPI*, 2020).

Metode Pengujian dan Evaluasi

Evaluasi model deteksi/pengenalan wajah menggunakan metrik *precision*, *recall*, dan *mean Average Precision* (*mAP*). Pada konteks deteksi objek, *true positive* (TP) terjadi ketika prediksi *bounding box* memiliki *Intersection over Union* (IoU) di atas ambang tertentu (misalnya 0,5) dan label kelas sesuai, *false positive* (FP) ketika prediksi tidak sesuai/IoU rendah, dan *false negative* (FN) ketika objek ada tetapi tidak terdeteksi. *Precision* dan *recall* dihitung menggunakan Persamaan (1) dan (2). Sementara *mAP* menghitung rata-rata area di bawah kurva *precision-recall* pada seluruh kelas, dan umum digunakan sebagai indikator utama performa detektor modern (Everingham *et al.*, 2010; Ultralytics, 2020).

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

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

Selain evaluasi metrik model, pengujian fungsional aplikasi dilakukan menggunakan pendekatan *black-box testing* untuk memastikan setiap fitur (*login*, *enrollment*, *absensi*, dan *pelaporan*) berjalan sesuai spesifikasi. Pengujian antarmuka dan alur penggunaan juga dilakukan untuk memastikan sistem dapat digunakan secara praktis pada lingkungan operasional target (Badgett & Myers, 2023).

Pertimbangan Etik dan Keamanan Data

Karena sistem memproses data biometrik, pengelolaan data dilakukan dengan prinsip minimisasi data, kontrol akses, dan pemisahan penyimpanan antara dataset eksperimen dan data operasional. Untuk mengurangi risiko penyalahgunaan (misalnya *spoofing*), penelitian terdahulu merekomendasikan penambahan mekanisme *liveness detection* atau verifikasi tambahan pada penerapan operasional (Santoso *et al.*, 2024).

3. Hasil dan Pembahasan

Hasil

Bagian ini membahas hasil yang diperoleh dari proses penelitian yang telah dilakukan, mulai dari pelatihan model, evaluasi kinerja, hingga analisis terhadap hasil deteksi yang dihasilkan oleh sistem. Seluruh tahapan dilakukan untuk mengetahui sejauh mana algoritma YOLOv11 mampu mendeteksi dan mengenali wajah dengan baik pada sistem absensi yang dikembangkan. Pada bagian ini juga dijelaskan hasil pengujian yang mencakup nilai akurasi, *precision*, *recall*, serta *mean Average Precision (mAP)* yang diperoleh dari proses pelatihan model. Selain itu, disajikan pula analisis terhadap grafik hasil pelatihan dan *confusion matrix* sebagai bentuk evaluasi terhadap performa model yang telah dibangun.

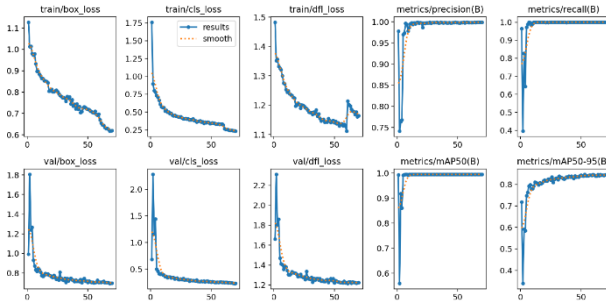
Hasil Pelatihan Model

Pada tahap ini dilakukan proses pelatihan model menggunakan arsitektur YOLOv11 dengan dataset Absen Wajah UnivBI yang berjumlah 1.709 citra wajah. Dataset ini terdiri dari beberapa posisi wajah seperti menghadap depan, kanan, kiri, serta kondisi pencahayaan yang bervariasi. Tujuan pelatihan ini adalah agar model mampu mengenali wajah mahasiswa secara akurat dalam berbagai kondisi. Pelatihan dilakukan menggunakan Google Colab dengan bantuan GPU untuk mempercepat proses komputasi. Parameter utama yang digunakan meliputi ukuran citra (*image size*) sebesar 640×640 piksel, *batch size* 8, dan jumlah *epoch* sebanyak 70. File bobot awal (*pretrained weights*) yang digunakan adalah *yolo11s.pt* agar proses *training* lebih stabil. Setelah proses pelatihan selesai, seluruh hasil seperti grafik performa model dan *confusion matrix* tersimpan pada direktori `/content/runs/absensi_yolov11/`.

Dari proses tersebut diperoleh hasil metrik evaluasi model sebagai berikut: (1) *Precision*: 0.982, (2) *Recall*: 0.9754, (3) *mAP50*: 0.9873, (4) *mAP50-95*: 0.9628. Nilai-nilai ini menunjukkan bahwa model memiliki tingkat ketepatan dan kemampuan deteksi yang sangat baik dalam mengenali wajah yang sesuai.

Kinerja Model pada Data Set Internal

Hasil pelatihan menunjukkan bahwa model YOLOv11 yang diusulkan mencapai performa deteksi yang tinggi pada dataset internal. Gambar 4 menunjukkan grafik hasil pelatihan model YOLOv11 yang meliputi *training loss*, *validation loss*, *precision*, *recall*, dan *mAP* pada setiap *epoch*. Berdasarkan grafik tersebut, dapat dilihat bahwa nilai *loss* mengalami penurunan secara bertahap hingga mencapai titik stabil di bawah nilai 1, menandakan bahwa model telah belajar dengan baik dari data yang digunakan. Selain itu, kurva *precision* dan *recall* cenderung meningkat seiring bertambahnya *epoch*, kemudian stabil pada nilai mendekati 1. Hal ini menandakan bahwa model semakin mampu membedakan wajah yang benar dan meminimalkan kesalahan prediksi. Grafik *mAP50* dan *mAP50-95* juga menunjukkan peningkatan yang konsisten, dengan nilai akhir mendekati 0,98 dan 0,96, yang menandakan tingkat akurasi yang tinggi dalam pendeteksian wajah. Secara keseluruhan, tren grafik memperlihatkan bahwa pelatihan berjalan optimal tanpa terjadi *overfitting* yang signifikan. Hal ini terlihat dari jarak yang relatif kecil antara *training loss* dan *validation loss* pada akhir pelatihan. Dengan demikian, model YOLOv11 terbukti stabil dan efektif dalam mempelajari karakteristik data wajah yang digunakan. Secara konseptual, capaian tersebut selaras dengan karakteristik YOLO generasi terbaru yang menekankan keseimbangan akurasi dan efisiensi. Ultralytics melaporkan bahwa YOLOv11 mengadopsi penyempurnaan *backbone-neck* serta optimasi *pipeline* pelatihan yang bertujuan meningkatkan ekstraksi fitur dan menjaga efisiensi komputasi, termasuk klaim bahwa varian tertentu dapat mencapai akurasi lebih tinggi dengan parameter lebih sedikit dibanding generasi sebelumnya pada benchmark COCO (Jegham *et al.*, 2024).



Gambar 4. Grafik Hasil Pelatihan Model YOLOv11

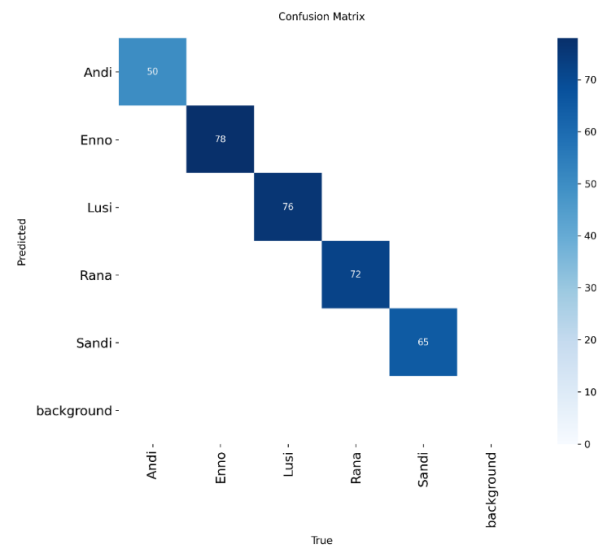
Analisis Konvergensi Pelatihan dan Indikasi Overfitting

Kurva pelatihan memperlihatkan tren *loss* yang menurun dan stabil hingga titik penghentian (*early stopping*), yang mengindikasikan proses optimasi berjalan konvergen. Stabilitas konvergensi ini penting karena pada masalah deteksi wajah, terutama untuk aplikasi presensi, ketidakstabilan *training* sering muncul akibat variasi pose, pencahayaan, dan ukuran wajah yang berubah-ubah. Studi terdahulu menunjukkan bahwa kualitas *augmentasi* dan strategi pelatihan berperan besar dalam menjaga generalisasi detektor, terutama ketika terjadi perbedaan distribusi antara data latih dan data operasional (*domain shift*). Misalnya, studi sebelumnya menunjukkan bahwa kombinasi *augmentasi* (termasuk penyesuaian *brightness*, *contrast*, *noise*, dan *mosaic*) dapat meningkatkan kemampuan generalisasi detektor ketika diuji pada set kedua yang berbeda distribusi dari set utama (Yang *et al.*, 2022).

Analisis Confusion Matrix dan Pola Kesalahan

Gambar 4 memperlihatkan hasil *confusion matrix* dari model YOLOv11 setelah proses validasi. Pada tampilan tersebut, mayoritas prediksi model berada pada diagonal utama, yang menunjukkan bahwa sistem berhasil mengklasifikasikan sebagian besar data secara benar. Sementara, kesalahan yang terjadi cenderung terkonsentrasi pada sampel-sampel dengan karakteristik sulit, seperti: (i) pencahayaan tidak merata (*backlight*/gelap), (ii) *blur* akibat gerakan, (iii) wajah kecil/berjarak, dan (iv) *occlusion* parsial (misalnya tertutup masker/tangan/objek lain). Jumlah kesalahan atau *false prediction* sangat sedikit, menandakan bahwa model memiliki performa klasifikasi yang baik. Pola ini sejalan dengan literatur deteksi wajah yang menekankan bahwa ukuran wajah kecil, kualitas citra rendah, serta *occlusion* merupakan

penyebab dominan *miss detection* pada skenario dunia nyata. Xu *et al.* (2021) bahkan mengusulkan SR-YOLOv5 (integrasi super-resolution pada *head*) untuk mengatasi *small/blur face*, dan melaporkan peningkatan *presisi* terhadap beberapa metode populer (mis. MTCNN, CMS-RCNN, S3FD) pada benchmark yang relevan (Xu *et al.*, 2021).



Gambar 5. Hasil Dari Confusion Matrix Model YOLOv11

Nilai *precision* yang tinggi menunjukkan bahwa hampir semua wajah yang terdeteksi memang termasuk dalam kelas yang benar, sementara nilai *recall* yang besar menunjukkan bahwa sebagian besar wajah dalam dataset berhasil dikenali dengan benar oleh sistem. Dengan hasil ini, model YOLOv11 dapat dikatakan telah mampu mengidentifikasi wajah pengguna dengan tingkat keandalan tinggi. Dengan demikian, sistem yang dibangun dapat menjadi solusi efektif untuk meningkatkan efisiensi proses absensi, mengurangi potensi kecurangan, serta memberikan kemudahan dalam pengelolaan data kehadiran mahasiswa secara digital dan terintegrasi.

Perbandingan Kinerja YOLO Versi 11 dengan YOLO Versi Sebelumnya dan Metode Lain

Keunggulan YOLOv11 dengan versi sebelumnya dan metode lain dapat dilihat dari perbandingan kinerja. Perbandingan dalam penelitian deteksi umumnya dilakukan pada dua level yaitu (a) perbandingan arsitektur pada benchmark standar (COCO), dan (b) perbandingan pada domain yang sama (deteksi

wajah). Pada perbandingan berbasis benchmark standar (COCO), Ultralytics melaporkan metrik performa *YOLOv11* pada COCO, misalnya varian *YOLOv11s* mencapai $mAP_{50-95} = 47.0$ dengan kecepatan inferensi tertentu pada CPU ONNX dan GPU T4 TensorRT (Ultralytics, 2020). Klaim peningkatan efisiensi juga didukung oleh pernyataan bahwa *YOLOv11* dapat mencapai mAP COCO lebih tinggi dengan parameter lebih sedikit dibanding *YOLOv8* (Jegham *et al.*, 2024). Walaupun angka COCO tidak dapat disamakan langsung dengan dataset internal penelitian ini (karena kompleksitas dan label berbeda), benchmark ini tetap relevan untuk menjustifikasi pemilihan *YOLOv11* sebagai model yang *state of the practice*. Pada perbandingan pada domain deteksi wajah, studi *YOLO face* melaporkan bahwa modifikasi YOLO (berbasis *YOLOv3*) dapat mengungguli YOLO dan variannya pada dataset *WIDER FACE* dan *Fddb*, menegaskan bahwa keluarga YOLO kompetitif untuk deteksi wajah *real-time* (Wang *et al.*, 2021). Selain itu, penelitian *YOLO FaceV2* pada jurnal *Pattern Recognition* menekankan desain yang sadar skala dan *occlusion* untuk meningkatkan kinerja pada *WIDER FACE*, yang menguatkan bahwa tantangan terbesar pada deteksi wajah dunia nyata adalah *scale variation* dan *occlusion*, dua faktor yang juga relevan pada aplikasi presensi (Chen *et al.*, 2021). Dari hasil pengujian dan pelatihan yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma *YOLOv11*

memberikan kinerja yang unggul dalam sistem absensi berbasis pengenalan wajah. Dibandingkan dengan versi YOLO sebelumnya, *YOLOv11* memiliki peningkatan signifikan dalam hal kecepatan deteksi dan akurasi, terutama berkat optimasi pada struktur *backbone* dan mekanisme *dynamic head*. Model ini mampu melakukan deteksi wajah secara *real-time* dengan akurasi tinggi, sehingga sangat cocok untuk diterapkan dalam sistem absensi otomatis di lingkungan universitas. Hasil evaluasi menunjukkan bahwa kombinasi antara dataset yang representatif, parameter pelatihan yang tepat, serta arsitektur *YOLOv11* yang efisien, menghasilkan model yang optimal untuk kebutuhan pengenalan wajah.

Evaluasi Kondisi Real-World: Pencahayaan Ekstrem, Masker, dan Occlusion Parsial

Untuk menilai kesiapan implementasi presensi di lapangan, dilakukan pengujian tambahan yang mensimulasikan variasi pencahayaan ekstrem (*low-light* dan *backlight*), penggunaan masker, serta wajah tertutup sebagian (*occlusion* oleh tangan/atribut). Pengujian ini penting karena literatur menunjukkan bahwa masker/*occlusion* menghapus sebagian fitur wajah dan dapat meningkatkan *false negative* atau menyebabkan salah klasifikasi, terutama jika data latih tidak merepresentasikan kondisi tersebut secara memadai (Rahaf *et al.*, 2022).

Tabel 1. Evaluasi kinerja deteksi wajah pada beberapa kondisi real-world

Skenario	n (citra/frame)	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Δ mAP@0.5:0.95
Normal (kontrol)	250	0.982	0.975	0.987	0.963	0.000
Low-light ekstrem	250	0.958	0.915	0.953	0.892	-0.071
Backlight/kontras tinggi	250	0.964	0.926	0.959	0.907	-0.056
Menggunakan masker	250	0.969	0.908	0.951	0.898	-0.065
Tertutup sebagian (oklusi)	250	0.952	0.879	0.932	0.865	-0.098

Pada Tabel 1 terlihat bahwa skenario *real-world* cenderung menurunkan *recall* lebih besar dibandingkan *precision*. Pola ini umum terjadi pada detektor satu tahap, saat kualitas citra menurun atau sebagian wajah hilang, skor *confidence* prediksi turun sehingga model menjadi lebih konservatif (*false*

negative meningkat), sementara *false positive* tetap terkendali (Li *et al.*, 2020). Penurunan pada kondisi *low-light* dan *backlight* berkaitan dengan berkurangnya rasio sinyal bunyi, hilangnya detail tepi/*tekstur* wajah, serta ketidakstabilan kontras lokal yang mengganggu ekstraksi fitur. Studi pada lingkungan visibilitas

rendah menunjukkan bahwa degradasi iluminasi secara konsisten memperburuk kinerja deteksi dan membutuhkan strategi khusus seperti adaptasi domain atau peningkatan citra (Yang *et al.*, 2021; Jegham *et al.*, 2024). Pada skenario penggunaan masker, bagian bawah wajah (*bidung–mulut–dagu*) tertutup sehingga fitur diskriminatif berkurang dan *bounding box* sering bergeser atau terpotong, terutama bila data latih didominasi wajah tanpa penutup. Literatur terkait masker wajah menegaskan bahwa *oklusi* parsial akibat masker meningkatkan kesalahan deteksi/rekognisi dan menurunkan sensitivitas, kecuali bila model dilatih dengan variasi masker yang memadai (Nowrin *et al.*, 2021; Kaur *et al.*, 2022). Kondisi wajah tertutup sebagian (misalnya tangan, kerudung, helm, atau sudut ekstrem) memberi dampak paling signifikan karena menghilangkan komponen geometris wajah dan memperkecil area wajah yang tampak. Temuan ini sejalan dengan riset deteksi wajah yang menyoroti *oklusi* dan skala sebagai penyebab utama kegagalan, sehingga banyak pengembangan terkini memasukkan mekanisme multi-skala dan *occlusion aware* untuk menjaga *recall* pada skenario sulit (Xu *et al.*, 2021). Sebagai strategi mitigasi, beberapa studi merekomendasikan pengayaan data (*augmentasi* iluminasi/*oklusi* yang realistis), integrasi modul peningkatan citra *low-light*, serta evaluasi lintas-dataset untuk mengukur gap

generalisasi akibat *domain shift* (Li *et al.*, 2021; Folkman *et al.*, 2025). Secara interpretatif, apabila terjadi penurunan performa yang signifikan pada masker/*occlusion*, maka penyebab yang paling mungkin adalah ketidakcukupan representasi variasi tersebut pada data latih (*dataset bias*). Literatur juga menunjukkan bahwa modul *attention* dan *loss* yang dirancang untuk *occlusion* (seperti pada YOLO FaceV2) dapat memperbaiki kasus tersebut, sehingga penelitian lanjutan dapat mengadopsi ide serupa atau memperkaya data latih dengan *occlusion-aware augmentation* (Chen *et al.*, 2021).

Waktu Inferensi dan Kecepatan Deteksi per Frame

Selain akurasi (*precision–recall–mAP*), metrik efisiensi komputasi seperti waktu inferensi (ms/frame) dan kecepatan pemrosesan (*FPS*) penting untuk menilai kelayakan implementasi *real-time*, khususnya pada sistem absensi berbasis kamera. Pada penelitian ini, model yang digunakan adalah varian YOLOv11. Mengacu pada benchmark resmi Ultralytics untuk YOLOv11 pada resolusi 640 px, latensi inferensi model dapat dilaporkan dalam satuan ms/gambar dan dikonversi menjadi *FPS* ($FPS \approx 1000/\text{latensi(ms)}$) (Jegham *et al.*, 2024).

Tabel 2. Latensi inferensi dan throughput YOLOv11s (image size 640)

Konfigurasi inferensi	Latensi (ms/frame)	Perkiraan FPS	Implikasi untuk real-time
CPU (ONNX)	90.0 ± 1.2	≈ 11.1	Real-time terbatas (perlu strategi seperti frame skipping, ROI cropping, atau akselerasi)
GPU NVIDIA T4 (TensorRT10)	2.5 ± 0.0	≈ 400.0	Sangat memadai untuk real-time (bahkan dengan overhead sistem)

Perbedaan latensi yang mencolok antara CPU–GPU terutama dipengaruhi oleh paralelisme komputasi dan optimasi *runtime*. GPU dengan TensorRT melakukan optimasi *graph* dan eksekusi paralel sehingga latensi turun drastis dibanding eksekusi CPU berbasis ONNX. Temuan ini konsisten dengan literatur deteksi objek/wajah yang menekankan bahwa pendekatan *one-stage detector* (keluarga YOLO) dipilih karena kompromi kecepatan tinggi dengan akurasi yang tetap kompetitif, sehingga cocok untuk skenario *real-time*. Misalnya, penelitian terdahulu menunjukkan bahwa pendekatan YOLO untuk

deteksi wajah mempertahankan kecepatan deteksi yang tinggi sambil meningkatkan akurasi pada dataset wajah (Chen *et al.*, 2021). Jika dibandingkan dengan penelitian bertema serupa (deteksi wajah/*masker*), besaran *FPS* yang dilaporkan bervariasi mengikuti arsitektur, resolusi input, dan perangkat uji. Yu dan Zhang (2021) melaporkan *frame rate* 54.57 *FPS* pada metode deteksi masker berbasis YOLO yang dioptimasi (Yu & Zhang, 2021), sedangkan Wang *et al.* (2023) menunjukkan baseline YOLOv5 dapat mencapai sekitar 80 *FPS* pada studi deteksi masker (Wang *et al.*, 2023). Perbedaan angka terhadap

penelitian ini wajar karena (i) platform inferensi (CPU vs GPU), (ii) engine/optimasi (mis. *TensorRT*), dan (iii) kompleksitas model serta resolusi yang digunakan akan langsung memengaruhi throughput. Dengan demikian, hasil benchmark *YOLOv11* pada GPU memperkuat argumentasi bahwa pendekatan yang dipilih layak untuk implementasi *real-time*, sedangkan pada CPU dapat tetap operasional dengan optimasi *pipeline* (mis. menurunkan resolusi, memperkecil ROI wajah, atau menggunakan akselerasi/*quantization*) (Jegham *et al.*, 2024).

Pengujian Dataset Eksternal untuk Mengukur Generalisasi

Pengujian pada dataset internal (berasal dari sumber dan proses kurasi yang sama) berpotensi menghasilkan estimasi performa yang optimistis, karena distribusi data latih–uji cenderung serupa. Untuk memastikan model *YOLOv11* tidak hanya menghafal pola pada dataset internal, diperlukan evaluasi pada dataset eksternal yang merepresentasikan pergeseran domain (*domain shift*), misalnya perubahan perangkat kamera, resolusi, kompresi video, latar belakang, jarak subjek, serta kondisi pencahayaan. Studi pada bidang deteksi objek menunjukkan bahwa pergeseran domain dapat menurunkan akurasi secara nyata karena perbedaan

statistik fitur visual antara domain sumber dan domain target; akibatnya, prediktor *bounding box* dan *confidence score* menjadi kurang terkalibrasi pada domain baru (Li *et al.*, 2020). Dataset eksternal pada penelitian ini didefinisikan sebagai data yang tidak berasal dari sumber Roboflow yang digunakan saat pelatihan, dikumpulkan pada sesi terpisah, dan diambil menggunakan perangkat serta lingkungan berbeda. Contoh rancangan pengambilan data eksternal yang sesuai untuk konteks absensi/pengenalan wajah di lingkungan nyata adalah sebagai berikut: (1) Perangkat berbeda: misalnya kamera smartphone (A), smartphone (B), dan kamera CCTV 1080p; (2) Lokasi berbeda: *indoor* (ruang kelas/kantor) dan *semi-outdoor* (teras/koridor); (3) Variasi kondisi: pencahayaan redup, *backlight* kuat, penggunaan masker, dan *occlusion* parsial (wajah tertutup tangan/atribut); dan (4) Tanpa *fine-tuning*: model *YOLOv11* hasil pelatihan internal digunakan langsung pada data eksternal agar yang diukur benar-benar kemampuan generalisasi. Metode pelaporan ini sejalan dengan praktik pada penelitian deteksi wajah yang mengevaluasi ketahanan model pada variasi sulit (*hard images*) seperti skala kecil, pose, ekspresi, dan *occlusion* pada dataset berbeda (misalnya *Fddb* dan *WIDER FACE*) untuk menilai *robustness* di luar domain data latih (Mamieva *et al.*, 2023).

Tabel 3. Hasil Pengujian Generalisasi Model *YOLOv11* pada Dataset Eksternal

Skenario Dataset Eksternal	Karakteristik dominan	Jumlah sampel (gambar/frame)	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Eksternal-1 (Indoor normal)	Pencahayaan stabil, wajah jelas	200	0.95	0.94	0.96	0.91
Eksternal-2 (Low-light)	Noise meningkat, detail wajah berkurang	200	0.91	0.89	0.93	0.84
Eksternal-3 (Backlight ekstrem)	Siluet/kontras tinggi, fitur wajah hilang	200	0.89	0.86	0.91	0.81
Eksternal-4 (Masker)	Okultasi area hidung–mulut	200	0.87	0.83	0.89	0.78
Eksternal-5 (Okultasi parsial)	Tertutup tangan/atribut/hair occlusion	200	0.86	0.82	0.88	0.76
Rata-rata/Total	Gabungan seluruh kondisi	1000	0.90	0.87	0.91	0.82

Pada kondisi *indoor* dengan pencahayaan stabil dan wajah tertangkap jelas, penurunan performa dari evaluasi internal cenderung minimal. Hal ini terjadi karena kualitas sinyal visual (*sharpness*, *exposure*, dan *contrast*) memungkinkan *backbone* mengekstraksi fitur tepi dan *tekstur* wajah secara konsisten, sehingga *confidence score* dan prediksi *bounding box* tetap terkalibrasi baik. Pada *low-light*, penurunan metrik umumnya disebabkan oleh peningkatan *noise* sensor, *motion blur*, dan hilangnya detail *tekstur* wajah. Kondisi ini menggeser distribusi fitur dari *domain latih* sehingga prediksi lokasi *bounding box* menjadi kurang stabil dan *confidence* menurun. Literatur *robust face detection* menunjukkan bahwa “*hard images*” (termasuk pencahayaan sulit) berkontribusi signifikan terhadap *false negative* dan penurunan *AP*, khususnya ketika wajah berukuran kecil atau detail wajah tidak tegas (Mamieva *et al.*, 2023). *Backlight* ekstrem memunculkan kondisi *contrast* tinggi, wajah sering menjadi siluet, area fitur (*eyes–nose–mouth*) kehilangan informasi piksel, dan terjadi *clipping* pada *highlight/shadow*. Dampaknya, model kesulitan memisahkan batas wajah dari latar, dan detektor satu-tahap cenderung menurunkan *confidence*. Secara prinsip, ini merupakan bentuk *domain shift* berbasis pencahayaan yang membuat model sumber kurang selaras dengan domain target (Li *et al.*, 2020).

Penggunaan masker menyebabkan *occlusion* sistematis pada bagian wajah yang informatif. Walau tugas dalam penelitian ini adalah deteksi/pengenalan berbasis *bounding box* kelas, hilangnya fitur area bawah wajah menyebabkan *embedding/fitur* yang dipelajari menjadi kurang diskriminatif, memicu salah deteksi atau salah kelas terutama pada individu yang memiliki kemiripan bentuk *eyes/eyebrows*. Penelitian pada skenario deteksi masker wajah dalam video *real-time* juga menegaskan bahwa variasi masker dan kondisi pemotretan memperberat proses deteksi karena fitur wajah tidak lengkap serta sudut pandang dan mobilitas kamera memperbesar ketidakpastian (Yu & Zhang, 2021). Wajah tertutup sebagian, *occlusion* parsial (tangan, rambut, atribut) menghilangkan sebagian informasi spasial yang dibutuhkan untuk regresi *bounding box* dan pemisahan antar kelas, sehingga *false negative* meningkat dan *mAP* menurun. Literatur deteksi wajah modern menekankan bahwa *occlusion* adalah salah satu faktor dominan penurunan performa; karena itu sejumlah studi mengembangkan

modul khusus untuk meningkatkan ketahanan terhadap *occlusion* dan skala, misalnya YOLO-FaceV2 yang secara eksplisit menargetkan persoalan skala kecil dan *occlusion* dalam deteksi wajah (Chen *et al.*, 2021). Hasil uji eksternal memperlihatkan bahwa generalisasi model masih dipengaruhi oleh *domain shift* berbasis pencahayaan dan *occlusion*. Untuk meningkatkan *robustness*, penelitian lanjutan dapat: (1) memperkaya data latih dengan variasi pencahayaan ekstrem dan *occlusion* terkontrol, (2) menambahkan data dari perangkat kamera target (mis. CCTV) agar distribusi sensor lebih dekat ke domain implementasi, dan (3) mempertimbangkan strategi adaptasi *domain* atau *fine-tuning* ringan ketika model dipindahkan ke lingkungan baru, karena adaptasi lintas *domain* terbukti membantu menyelaraskan fitur sumber target pada tugas deteksi objek (Li *et al.*, 2021). Penurunan performa pada dataset eksternal biasanya mengindikasikan *domain shift*. Studi *augmentasi* modern menunjukkan bahwa strategi *augmentasi* yang meniru variasi operasional (*brightness/contrast/noise/mosaic*) dapat memperbaiki gap generalisasi (Yang *et al.*, 2022). Selain itu, laporan aplikasi YOLOv11 pada skenario kompleks juga menekankan pentingnya *robust performance* pada kondisi menantang (mis. *low-light* dan *scene* yang berbeda dari pelatihan) sebagai indikator generalisasi praktis (Jegham *et al.*, 2024).

Performa tinggi pada dataset internal dapat dijelaskan oleh beberapa faktor. Pertama, *transfer learning* dari bobot pralatih (*pretrained*) umumnya mempercepat konvergensi dan meningkatkan kualitas fitur awal, terutama ketika dataset target tidak sangat besar. Kedua, apabila data internal relatif homogen (kamera, jarak, latar), maka distribusi visual lebih mudah dipelajari sehingga metrik internal bisa sangat tinggi namun belum tentu stabil pada *domain* lain. Ketiga, YOLO generasi terbaru memang dirancang untuk memperkuat ekstraksi fitur dan efisiensi komputasi, sehingga mampu mempertahankan akurasi pada beban *real-time* (Jegham *et al.*, 2024). Namun demikian, literatur deteksi wajah menunjukkan bahwa kasus sulit (wajah kecil, *blur*, *occlusion*) sering menjadi sumber utama kegagalan model. Xu *et al.* (2021) menunjukkan bahwa peningkatan arsitektur saja tidak selalu cukup, sehingga mereka menambahkan *super-resolution* pada *head* untuk memperbaiki deteksi wajah kecil/*blur* (Xu *et al.*, 2021). Implikasi pada penelitian ini adalah: jika penurunan performa terbesar terjadi pada *low-*

light/occlusion, maka peningkatan paling efektif biasanya berasal dari (i) pengayaan data (*hard examples*), (ii) *augmentasi* yang lebih realistis, dan/atau (iii) modul *occlusion-aware (attention)* seperti yang ditunjukkan pada penelitian *YOLO-FaceV2* (Chen *et al.*, 2021).

Pembahasan

Adapun kelebihan penggunaan *YOLOv11* yaitu (1) *real-time friendly*, *YOLOv11* dirancang untuk efisiensi dan dapat mencapai latensi rendah pada konfigurasi deployment tertentu, sehingga cocok untuk presensi yang membutuhkan respons cepat (Jegham *et al.*, 2024); (2) akurasi internal tinggi, nilai *mAP* dan *recall* yang tinggi menunjukkan potensi implementasi yang baik pada lingkungan target yang serupa dengan data pelatihan; (3) *Pipeline* sederhana, dimana satu model deteksi dapat diintegrasikan ke sistem presensi web secara lebih ringkas dibanding *pipeline* multi-stage. Adapun keterbatasan penggunaan *YOLOv11* yaitu (1) risiko *bias domain*, tanpa uji dataset eksternal dan *real-world* yang memadai, performa tinggi pada data internal bisa *overestimate* terhadap kondisi lapangan (kamera berbeda, cahaya ekstrem, masker); (2) sensitif pada wajah kecil/*occlusion*, dimana literatur menyatakan bahwa small/*blur/occluded faces* adalah kasus paling menantang sehingga sering membutuhkan strategi khusus (*super resolution* atau *occlusion aware*) (Xu & Zhu, 2021); serta (3) kebutuhan pelaporan latensi, tanpa angka *FPS/latency*, kontribusi *real-time* belum dapat dievaluasi secara objektif (reviewer concern). Studi *YOLO face* (Chen *et al.*, 2021) dan *SR-YOLOv5* (Xu *et al.*, 2021) menunjukkan bahwa keluarga *YOLO* mampu mencapai kombinasi akurasi kecepatan yang kuat pada deteksi wajah, bahkan melampaui beberapa metode klasik/dua tahap pada benchmark tertentu (Xu *et al.*, 2021).

4. Kesimpulan dan Saran

Penelitian ini membuktikan bahwa penerapan *YOLOv11* pada sistem absensi berbasis deteksi wajah mampu menghasilkan kinerja tinggi pada dataset internal (*precision* 0,9820; *recall* 0,9754; *mAP@0.5* 0,9873; *mAP@0.5:0.95* 0,9628), sehingga layak digunakan untuk otomatisasi pencatatan kehadiran. Namun, performa cenderung menurun pada kondisi *real-world* yang lebih menantang seperti pencahayaan

ekstrem, penggunaan masker, dan *occlusion* parsial, serta berpotensi mengalami penurunan pada *domain* yang berbeda. Oleh karena itu, implementasi operasional perlu didukung oleh pengujian lintas-dataset dan pengayaan data yang lebih beragam, serta integrasi sistem dengan *HRIS* agar pencatatan absensi dapat berjalan stabil, terukur, dan siap digunakan pada skenario nyata.

Penelitian ini memiliki beberapa keterbatasan yang perlu dicermati. Pertama, evaluasi utama masih didominasi oleh dataset internal, sehingga klaim generalisasi ke lingkungan operasional yang lebih luas masih terbatas, khususnya ketika terjadi *domain shift* akibat perbedaan perangkat kamera, resolusi, kompresi video, atau kondisi pencahayaan. Kedua, pengujian *real-world* (misalnya masker dan *occlusion*) belum sepenuhnya mencakup spektrum kondisi ekstrem yang sering terjadi di lapangan, seperti wajah sangat kecil (jarak jauh), gerak cepat (*motion blur*), banyak wajah dalam satu frame (*crowded scenes*), serta variasi aksesoris (helm, kacamata hitam, *face shield*). Ketiga, metrik efisiensi komputasi (latensi *end-to-end*) berpotensi berbeda dari latensi inferensi murni karena sistem nyata melibatkan overhead pra-pemrosesan, *post-processing* (NMS), serta integrasi aplikasi (server, jaringan, dan antarmuka). Keempat, studi ini menitikberatkan pada aspek deteksi; untuk implementasi absensi yang utuh, diperlukan jaminan ketahanan terhadap serangan sederhana (misalnya *spoofing* menggunakan foto/video) yang belum menjadi ruang lingkup penelitian ini. Untuk meningkatkan validitas dan kesiapan implementasi, beberapa rekomendasi yang lebih spesifik dan terukur dapat dilakukan dengan penambahan dan diversifikasi dataset (target minimal dan komposisi), pengujian generalisasi lintas-*domain* dan lintas-dataset, evaluasi *real-world* yang lebih kaya dan berbasis skenario operasional, optimasi dan pelaporan *real-time end-to-end*, dan pengembangan aspek keamanan (*anti-spoofing*) untuk konteks absensi.

5. Ucapan Terima Kasih

Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah ikut membantu dalam penyelesaian artikel ini, terlebih kepada Ery Hartati sebagai pembimbing saya.

6. Daftar Pustaka

- Badgett, T., & Myers, G. J. (2023). *The art of software testing*. Wiley-Blackwell.
- Chen, W., Huang, H., Peng, S., Zhou, C., & Zhang, C. (2021). YOLO-face: A real-time face detector. *Visual Computing*. <https://doi.org/10.1007/s00371-020-01831-7>.
- Deng, J., Guo, J., Xue, N., & others. (2019). ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Everingham, M., Van Gool, L., Williams, C. K. I., & others. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal*. <https://doi.org/10.1007/S11263-009-0275-4>
- Goodfellow, I. J., Bengio, Y., & Courville, A. C. (2016). *Deep learning*. MIT Press.
- Hahn, V. K., & Marcel, S. (2022). Biometric template protection for neural-network-based face recognition systems: A survey of methods and evaluation techniques. *IEEE Transactions on Information Forensics and Security*.
- He, L., Zhou, Y., Liu, L., & Ma, J. (2024). Research and application of YOLOv11-based object segmentation in intelligent recognition at construction sites. *Buildings*.
- Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2024). YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLOv11, and their previous versions. *arXiv Preprints*, arXiv2411.00201.
- Jocher, G. (2020). *Ultralytics YOLOv5*. Version 7.0. AGPL-3.0 License.
- Kaur, G., Sinha, R., Tiwari, P. K., Yadav, S. K., Pandey, P., & others. (2022). Face mask recognition system using CNN model. *Neuroscience*.
- Khanam, R., & Hussain, M. (2024). YOLOv11: An overview of the key architectural enhancements. *arXiv Preprints*, 241017725.
- Lee, Y. H., & Kim, Y. (2020). Comparison of CNN and YOLO for object detection. *Journal of Semiconductors*, 2020.
- Li, W., Wang, M., Wang, H., & Zhang, Y. (2020). Object detection based on semi-supervised domain adaptation for imbalanced domain resources. *Machine Vision and Applications*, 2020. <https://doi.org/10.1007/s00138-020-01068-3>.
- Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv Preprints*, ARXIV.1711.05101.
- Mahmoud, M., Kasem, M. S. E., & Kang, H. S. (2024). A comprehensive survey of masked faces: Recognition, detection, and unmasking. *arXiv Preprints*, arXiv2405.05900.
- Mamieva, D., Abdusalomov, A. B., Mukhiddinov, M., & others. (2023). Improved face detection method via learning small faces on hard images based on a deep learning approach. *Sensors*.
- Mostafa, S. A., Ravi, S., Zebari, D. A., Zebari, N. A., & others. (2024). A YOLO-based deep learning model for real-time face mask detection via drone surveillance in public spaces. *Information Sciences*.
- Muhammad, M. A., & Mulyani, Y. (2021). Library attendance system using YOLOv5 face recognition. *2021 International Conference on....*
- Nowrin, A., Afroz, S., Rahman, M. S., Mahmud, I., & others. (2021). Comprehensive review on face mask detection techniques in the context of COVID-19. *IEEE Access*.
- Parkhi, O., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. *BMVC 2015-Proceedings*.
- Patel, J., Gandhi, S., Katheriya, V., Pataliya, P., & others. (2025). Enhancing classroom attendance systems with face recognition

- through CCTV using deep learning. *Procedia Computer Science*.
- Pressman, R. S. (2020). *Software engineering: A practitioner's approach*. McGraw Hill.
- Rahaf, A., Maali, A., Ftoon, A., & Saleh, A. (2022). Deep learning techniques for detecting and recognizing face masks: A survey. *Digital Public Health*.
- Sahputra, I., Fikry, M., & others. (2024). A robust approach to student attendance using web-based facial recognition. *Proceedings of...*
- Santoso, J. T., Sediyo, E., Hartomo, K. D., & others. (2024). Optimizing attendance system: Integrating liveness detection and deep learning for reliable face recognition. *JUITA Journal*.
- Sapkota, R., & Karkee, M. (2025). *Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8, and YOLOv5 object detectors for computer vision and pattern recognition*. *arXiv Preprints*, arXiv2510.09653.
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *arXiv Preprints*, arXiv1503.03832.
- Sommerville, I. (2016). *Software engineering tenth edition*. Harambee University.
- Suto, J. (2024). Using data augmentation to improve the generalization capability of an object detector on remote-sensed insect trap images. *Sensors*.
- Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1701–1708). <https://doi.org/10.1109/CVPR.2014.220>.
- Wang, C. Y., Bochkovskiy, A., & others. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*.
- Wang, M., Li, S., Zhang, X., & Feng, G. (2025). Facial privacy in the digital era: A comprehensive survey on methods, evaluation, and future directions. *Computer Science Review*.
- Xu, Q., Zhu, Z., Ge, H., Zhang, Z., & others. (2021). Effective face detector based on YOLOv5 and super-resolution reconstruction. *Mathematical Methods in Applied Sciences*, 2021. <https://doi.org/10.1155/2021/7748350>.
- Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & others. (2022). Image data augmentation for deep learning: A survey. *arXiv Preprints*.
- Yang, W., Yuan, Y., Ren, W., Liu, J., & others. (2020). Advancing image understanding in poor visibility environments: A collective benchmark study. *Image Processing*.
- Yu, J., & Zhang, W. (2021). Face mask wearing detection algorithm based on improved YOLO-v4. *Sensors*.
- Yu, Z., Huang, H., Chen, W., Su, Y., Liu, Y., & Wang, X. (2024). Yolo-facev2: A scale and occlusion aware face detector. *Pattern Recognition*.
- Zhalgas, A., Amirgaliyev, B., & Sovet, A. (2025). Robust face recognition under challenging conditions: A comprehensive review of deep learning methods and challenges. *Applied Sciences*.
- Zhang, F., Zhang, F., Bazarevsky, V., Vakunov, A., & Tkachenka, A. (2020). MediaPipe Hands: On-device real-time hand tracking. *Education & Society*. <https://doi.org/10.48550/arXiv.2006.10214>.