

# Pengembangan Aplikasi Presensi Berbasis Deep Learning

*Development of a Deep Learning Based Attendance Application*

Lailatul Akmal<sup>1</sup>, Ilman Zuhri Yadi<sup>2</sup>, Yesi Novaria Kunang<sup>3</sup>, Fatma Sari<sup>4</sup>  
Sistem Informasi, Sains Teknologi/Universitas Bina Darma  
Palembang, Sumatera Selatan, Indonesia

E-mail: [1lailatulprabumulih@gmail.com](mailto:lailatulprabumulih@gmail.com), [2ilmanzuhriyadi@binadarma.ac.id](mailto:ilmanzuhriyadi@binadarma.ac.id),  
[3yesinovariakunang@binadarma.ac.id](mailto:yesinovariakunang@binadarma.ac.id), [4fatmasari@binadarma.ac.id](mailto:fatmasari@binadarma.ac.id)

## Abstrak

Sistem presensi berbasis pengenalan wajah menjadi solusi modern untuk mengatasi kelemahan metode presensi manual yang rawan manipulasi dan kesalahan pencatatan. Penelitian ini mengembangkan aplikasi presensi berbasis deep learning dengan menerapkan Convolutional Neural Network (CNN) menggunakan arsitektur MobileNetV2, VGG16, dan ResNet50 yang dioptimalkan untuk perangkat dengan sumber daya terbatas. Dataset wajah dikumpulkan secara mandiri dan melalui tahapan prapemrosesan, termasuk normalisasi, resizing, augmentasi, serta deteksi wajah dengan OpenCV. Model dilatih menggunakan TensorFlow dan Keras pada Google Colab dengan GPU. Lalu dievaluasi menggunakan confusion matrix yang hasil prediksinya akurat dan tingkat kesalahannya yang rendah. Serta dilakukan evaluasi melalui classification report yang terdiri dari akurasi yang mencapai 0.98, precision 1.00, recall 1.00, dan F1-score 1.00 yang mencapai tingkat performa yang sangat tinggi sehingga menandakan bahwa tidak ada kesalahan prediksi yang terjadi. Aplikasi web berbasis Flask dirancang untuk menghubungkan model pengenalan wajah dengan antarmuka pengguna, serta diuji secara real-time untuk mengukur kecepatan dan akurasi presensi. Hasil penelitian menunjukkan bahwa aplikasi presensi berbasis CNN mampu memberikan alternatif presensi yang lebih aman, cepat, dan efisien dibandingkan metode konvensional.

**Kata Kunci:** Deep Learning, CNN, Face Recognition, Presensi, MobileNetV2

## Abstract

A facial recognition-based attendance system is a modern solution to overcome the weaknesses of manual attendance methods that are prone to manipulation and recording errors. This study uses a deep learning-based attendance application by implementing a Convolutional Neural Network (CNN) using MobileNetV2, VGG16, and ResNet50 architectures optimized for devices with limited resources. The facial dataset was collected independently and went through preprocessing stages, including normalization, resizing, augmentation, and face detection with OpenCV. The model was trained using TensorFlow and Keras on Google Colab with a GPU. It was then evaluated using a confusion matrix, which yielded accurate predictions with a low error rate. A classification report was also conducted, with an accuracy of 0.98, a precision of 1.00, a recall of 1.00, and an F1-score of 1.00, achieving a very high level of performance, indicating no prediction errors. A Flask web-based application was designed to connect the facial recognition model with the user interface, and was tested in real-time to measure the speed and accuracy of attendance. The results show that the CNN-based attendance application is able to provide a safer, faster, and more efficient attendance alternative compared to conventional methods.

**Keywords:** Deep Learning, CNN, Face Recognition, Presence, MobileNetV2

## 1. PENDAHULUAN

Presensi merupakan aspek penting dalam manajemen kehadiran di berbagai institusi, baik pendidikan, perusahaan, maupun organisasi. Metode presensi konvensional seperti tanda tangan manual atau kartu identitas masih memiliki banyak kelemahan, antara lain rawan manipulasi, pencatatan yang tidak akurat, serta kurang efisien dalam penggunaan waktu. Untuk mengatasi permasalahan tersebut, teknologi pengenalan wajah berbasis kecerdasan buatan mulai dikembangkan sebagai alternatif sistem presensi yang lebih aman dan efisien.

*Convolutional Neural Network (CNN)* merupakan salah satu arsitektur *deep learning* yang banyak digunakan dalam pengenalan wajah. *CNN* mampu mengekstraksi fitur wajah secara otomatis dan telah terbukti memberikan akurasi tinggi dalam proses identifikasi. Sejumlah penelitian menunjukkan bahwa *CNN* dapat mencapai akurasi di atas 90% pada sistem presensi otomatis, jauh melampaui metode berbasis deteksi manual. Namun, tantangan tetap ada, terutama terkait keterbatasan sumber daya komputasi, kondisi pencahayaan, variasi ekspresi, serta kebutuhan akan privasi data biometrik.

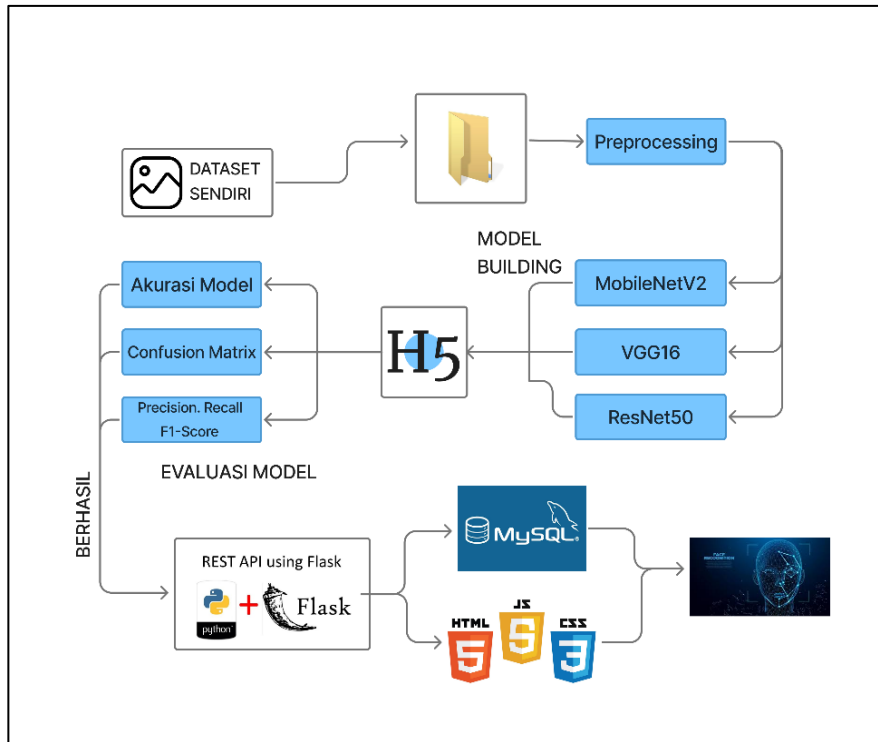
Untuk menjawab tantangan tersebut, penelitian ini mengembangkan aplikasi presensi berbasis *CNN* dengan membandingkan tiga arsitektur populer, yaitu *MobileNetV2*, *VGG16*, dan *ResNet50*. *MobileNetV2* dipilih karena efisiensinya pada perangkat dengan sumber daya terbatas, sementara *VGG16* dan *ResNet50* digunakan sebagai pembanding untuk mengevaluasi kinerja model. Dataset wajah dikumpulkan secara mandiri, kemudian melalui tahap prapemrosesan berupa normalisasi, *resizing*, augmentasi, dan deteksi wajah menggunakan *OpenCV*.

Aplikasi presensi dikembangkan berbasis web menggunakan *Flask*, yang berfungsi sebagai *backend* untuk mengintegrasikan model pengenalan wajah dengan antarmuka pengguna. Dengan desain ini, sistem dapat dijalankan secara *real-time* melalui browser desktop maupun perangkat mobile tanpa perlu aplikasi tambahan.

## 2. METODOLOGI

### 2.1 Alur Penelitian

Alur penelitian ini menggambarkan tahapan utama dalam pengembangan pengenalan wajah berbasis *CNN*. Proses ini mencakup pengumpulan data, *preprocessing*, pembangunan model, evaluasi model, serta implementasi dalam aplikasi web menggunakan *Flask* dengan integrasi basis data *MySQL* dan antarmuka *HTML*, *CSS*, dan *JavaScript*. Tahapan ini dirancang agar sistem dapat memahami, mengolah, serta memberikan respons yang akurat terhadap wajah yang dideteksi. Sebagai ilustrasi, Gambar 1. Alur Penelitian berikut menjelaskan hubungan antara tiap tahapan dalam proses penelitian.



Gambar 1. Alur penelitian

## 2.2 Tahapan Penelitian

### 2.2.1 Pengumpulan Data

Penelitian ini dimulai dengan proses pengumpulan *dataset* yang berasal dari satu sumber utama, yaitu *dataset* yang dikumpulkan secara mandiri. Setelah data terkumpul, *dataset* dibagi menjadi dua bagian, yaitu data training (80%) dan data testing (20%) yang akan digunakan untuk pelatihan serta evaluasi model. *Dataset* gambar wajah ini mencakup berbagai variasi kondisi pencahayaan, sudut pengambilan, dan ekspresi wajah yang berbeda guna meningkatkan keandalan model dalam mengenali wajah secara akurat. Tahap ini juga bertujuan untuk memastikan kualitas data sebelum memasuki proses *preprocessing*, sehingga model dapat bekerja lebih optimal dalam mendeteksi serta mengklasifikasikan wajah.

### 2.2.2 Preprocessing Data

Sebelum digunakan dalam pelatihan model, *dataset* gambar wajah melalui beberapa tahapan *preprocessing* untuk memastikan bahwa data memiliki kualitas yang baik dan dapat diproses secara optimal oleh model *deep learning*. Tahapan *preprocessing* yang dilakukan meliputi:

1. Deteksi dan Pemotongan Wajah → Menggunakan *Haar Cascade* dari *OpenCV* untuk mendeteksi serta memotong area wajah, sehingga model lebih fokus pada fitur wajah dan tidak terganggu oleh latar belakang.

2. *Resizing* → Mengubah ukuran citra hasil cropping menjadi 224×224 piksel agar sesuai dengan format input arsitektur *MobileNetV2*, *VGG16*, dan *ResNet50*.
3. Normalisasi → Mengonversi nilai piksel citra ke dalam rentang 0–1, sehingga mempercepat konvergensi model dan meningkatkan stabilitas saat pelatihan.
4. Augmentasi Data → Memberikan variasi tambahan pada dataset melalui transformasi seperti rotasi, *flipping* horizontal, penyesuaian tingkat kecerahan (*brightness adjustment*), serta zooming. Teknik ini bertujuan memperkaya keragaman data dan mengurangi risiko *overfitting*.

### 2.2.3 Model Building

Pada tahap ini, membangun *model Convolutional Neural Network (CNN)* berbasis *deep learning* yang digunakan untuk pengenalan wajah. Arsitektur model yang digunakan dalam penelitian ini meliputi:

1. *MobileNetV2* → Model yang ringan dan efisien, dirancang untuk perangkat dengan keterbatasan sumber daya komputasi.
2. *VGG16* → Model dengan struktur *deep CNN*, yang memiliki tingkat akurasi tinggi dalam tugas klasifikasi gambar.
3. *ResNet50* → Model dengan pendekatan *residual learning*, yang dapat menangani jaringan yang lebih dalam tanpa mengalami masalah *vanishing gradient*.

Setelah model dilatih menggunakan dataset hasil *preprocessing*, model terbaik disimpan dalam format *H5 (.h5)* agar dapat digunakan dalam tahap implementasi selanjutnya.

### 2.2.4 Evaluasi Model

Setelah proses pelatihan selesai, model diuji dan dievaluasi menggunakan berbagai metrik untuk memastikan performanya dalam mengenali wajah dengan akurasi tinggi. Evaluasi dilakukan melalui beberapa tahap berikut:

1. Akurasi Model → Mengukur tingkat keberhasilan model dalam mengenali wajah menggunakan data training dan testing.
2. *Confusion Matrix* → Menilai kinerja model dalam mengklasifikasikan wajah dengan benar serta mendeteksi kemungkinan kesalahan klasifikasi.
3. *Precision*, *Recall*, dan *F1-Score* → Mengukur tingkat keakuratan model dalam mengenali wajah berdasarkan hasil klasifikasi yang benar dan salah.

Jika hasil evaluasi menunjukkan bahwa model belum mencapai akurasi optimal, maka dilakukan penyesuaian *hyperparameter* serta *augmentasi* data tambahan hingga model menunjukkan performa yang diinginkan.

### 2.2.4 Implementasi REST API dengan Flask

Sebelum implementasi, telah dilakukan pemilihan model guna memperoleh performa terbaik dalam proses evaluasi model. Model dengan hasil terbaik dari tahap evaluasi kemudian dipilih untuk diintegrasikan dan diterapkan dalam *REST API* menggunakan *Flask*.

### 2.2.5 Integrasi dengan Aplikasi Web

*REST API* berbasis *Flask* kemudian dihubungkan dengan aplikasi web menggunakan *HTML*, *CSS*, dan *JavaScript* yang bertindak sebagai antarmuka pengguna. Melalui aplikasi web ini, pengguna dapat melakukan absensi secara *real-time* melalui browser, sementara data hasil pengenalan wajah dicatat dalam basis data *MySQL*.

### 2.2.6 Pengujian berbasis kamera web

Tahap akhir dalam penelitian ini adalah pengujian aplikasi web untuk menyalakan kinerja sistem dalam kondisi nyata. Pengujian dilakukan dengan beberapa skenario berikut:

1. Pengujian dengan Data Wajah Baru → Menggunakan wajah yang tidak termasuk dalam pelatihan dataset.
2. Pengujian Performa *Real-Time* melalui *Webcam* → Mendeteksi wajah dengan kondisi pencahayaan, sudut pengambilan, dan ekspresi berbeda.
3. Analisis Kecepatan dan Respons Sistem → Mengukur kecepatan deteksi wajah, pengiriman data ke server *Flask*, proses inferensi model, penyimpanan hasil ke database *MySQL*, serta respon ke antarmuka web.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari gambar wajah yang dikumpulkan secara mandiri. Dataset ini dibagi menjadi dua bagian utama: data training, dan data testing.

```
Ahmad Dimas Aldian Alifurqon => TRAIN: +34 | TEST: +9
Andi Pramono => TRAIN: +8 | TEST: +2
Andi Septiyani => TRAIN: +13 | TEST: +4
Andryansah => TRAIN: +5 | TEST: +2
Annisah => TRAIN: +8 | TEST: +2
Asep Nurhikmah => TRAIN: +36 | TEST: +18
Attasyah Ramadhan => TRAIN: +13 | TEST: +4
Delwa Petrawan => TRAIN: +8 | TEST: +2
Deni Trilisandi => TRAIN: +8 | TEST: +2
Dini Febriyanti => TRAIN: +8 | TEST: +2
Dwi Alfiansyah => TRAIN: +14 | TEST: +4
Fathur Ramadhan => TRAIN: +13 | TEST: +4
Firdaus => TRAIN: +8 | TEST: +2
Jenny => TRAIN: +28 | TEST: +5
Jodi Akbar => TRAIN: +14 | TEST: +4
Kadek Sarasitika => TRAIN: +8 | TEST: +2
Kirana Naya Sari => TRAIN: +8 | TEST: +2
Lailatul Akmal => TRAIN: +72 | TEST: +18
Lipin => TRAIN: +12 | TEST: +3
Liza Riska Yambeyabdi => TRAIN: +17 | TEST: +5
Maryani Farwati => TRAIN: +8 | TEST: +2
Muhammad Risdianto Utomo => TRAIN: +15 | TEST: +4
Muhammad Rizki Safran => TRAIN: +8 | TEST: +2
Nadya Sukma Andini => TRAIN: +8 | TEST: +2
Nafari => TRAIN: +12 | TEST: +4
Naufal Rahmatullah => TRAIN: +14 | TEST: +4
Ostrinedia Rahmaga => TRAIN: +8 | TEST: +2
Patricia Aprilia Quiroz => TRAIN: +11 | TEST: +3
Riski Ardianto => TRAIN: +8 | TEST: +2
Zaskia Putri => TRAIN: +8 | TEST: +2
indi.eldiantika => TRAIN: +11 | TEST: +3
zakiyah => TRAIN: +5 | TEST: +2
zhorifroh khoirunnisa => TRAIN: +8 | TEST: +2
=====
Selesai. Total disalin => TRAIN: 451 | TEST: 122
Folder tanpa gambar valid: 0
=====
```

Gambar 2. Hasil pembagian dataset train & test

Setelah dataset diperoleh, dilakukan proses seleksi dan pembagian gambar ke dalam kategori yang sesuai. Data ini kemudian diklasifikasikan ke dalam dua file utama, yaitu training 80% dan testing 20%, untuk memastikan model dapat melakukan generalisasi dengan baik. Berdasarkan analisis awal, jumlah gambar dalam dataset yang dikumpulkan adalah 573 gambar yang terdiri dari 33 kelas yang berbeda dan gambar dalam dataset yang diambil secara mandiri.

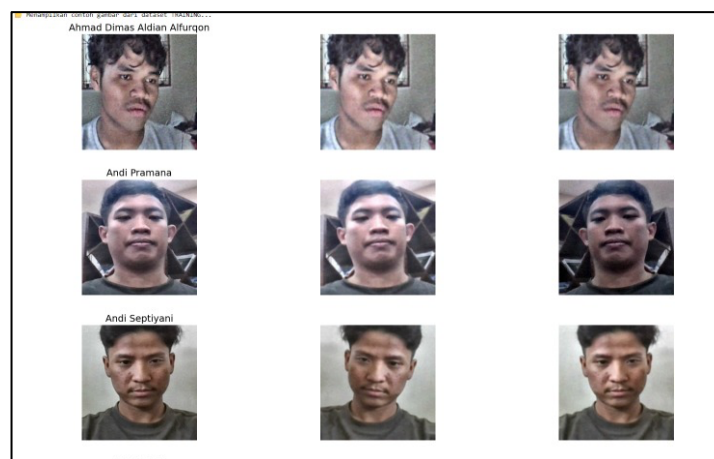
### 3.2 Preprocessing Data

Tahap prapemrosesan dataset merupakan langkah awal dilakukan untuk meningkatkan kualitas gambar sebelum digunakan dalam pelatihan model. Pada tahap ini, gambar wajah yang berasal dari dataset mentah melalui serangkaian proses mulai dari deteksi wajah menggunakan *Haar Cascade*, pemotongan (crop) area wajah beserta leher dengan margin tertentu, hingga penyesuaian ukuran citra agar seragam. Selanjutnya dilakukan normalisasi pencahayaan dengan metode *CLAHE* untuk menyeimbangkan kondisi terang-gelap, serta pengecekan kualitas citra berdasarkan tingkat ketajaman dan kecerahan sehingga gambar yang buram atau terlalu gelap/terang dapat dihindari. Untuk memperkaya variasi data latih, diterapkan *augmentasi* berupa *rotasi*, *flipping*, perubahan *brightness* dan *contrast*, penambahan *noise*, maupun *blur*, sehingga model lebih robust terhadap kondisi nyata. Hasil akhir dari tahap ini berupa dataset wajah+leher yang telah distandarisasi dan disimpan sesuai struktur train dan test, lengkap dengan metadata kelas dan konfigurasi prapemrosesan agar dapat digunakan secara konsisten pada proses pelatihan selanjutnya.

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).  
Selesai.  
Train + saved: 304 aug: 608 skip: 147 reasons: {'no_face': 90, 'blur': 57}  
Test + saved: 82 aug: 0 skip: 40 reasons: {'no_face': 20, 'blur': 20}
```

Gambar 3. Output data train & test

Hasil dari prapemrosesan ini kemudian disimpan ke direktori keluaran sesuai struktur train dan test, dilengkapi dengan metadata berupa daftar kelas dan konfigurasi prapemrosesan agar eksperimen dapat direplikasi secara konsisten.



#### Gambar 4. Hasil menampilkan beberapa gambar dari setiap folder

Pada Gambar 4 menampilkan sejumlah gambar dari setiap subfolder yang mewakili kelas tertentu (data mandiri) di dalam direktori dataset yang telah dilakukan augmentasi. Setiap subfolder berisi gambar wajah dari satu kelas, dan gambar-gambar tersebut ditampilkan secara berurutan dalam grid yang terorganisir. Proses visualisasi ini merupakan langkah awal yang sangat penting sebelum pelatihan model. Berdasarkan hasilnya, kita dapat memastikan bahwa dataset memiliki gambar berkualitas, bervariasi, dan relevan untuk pengenalan wajah, sehingga memberikan dasar yang kuat bagi model untuk belajar dengan efektif.

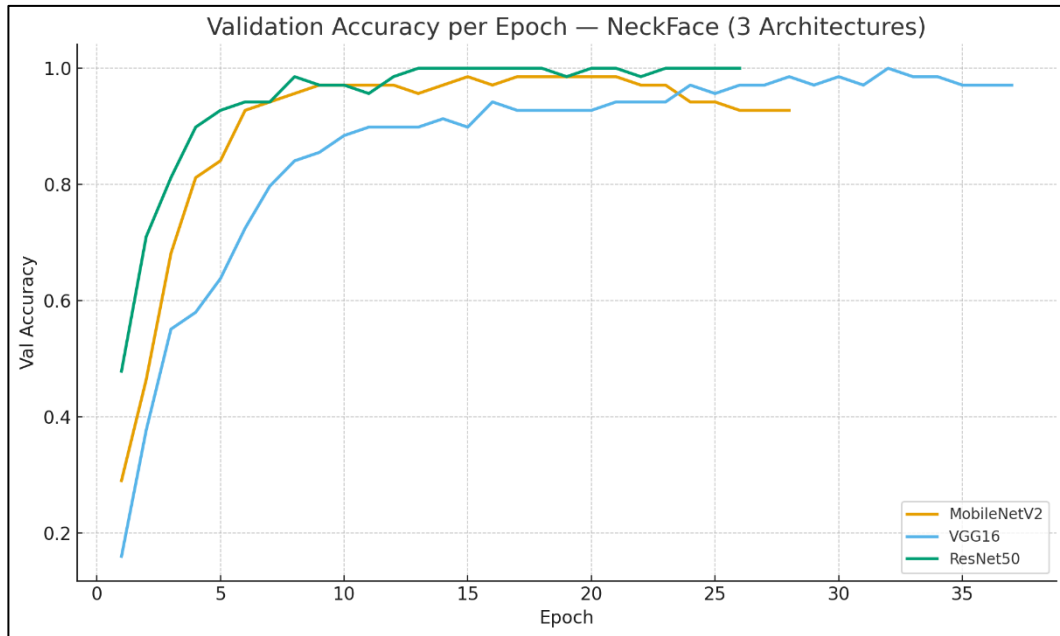
### 3.3 Model Building

*Model building* atau pelatihan model dilakukan dengan menggunakan tiga konfigurasi arsitektur jaringan konvolusional, yaitu *MobileNetV2*, *VGG16*, dan *ResNet50*. Proses pelatihan dilakukan dengan menggunakan dataset wajah yang dikumpulkan secara mandiri. Tujuannya adalah membangun model pengenalan wajah yang akurat terhadap berbagai variasi kondisi, seperti pencahayaan, dan kualitas citra.

Parameter pelatihan yang digunakan pada ketiga model:

1. *Epochs*: 30 (ditambah *fine-tuning* 15 *epoch* tambahan)
2. *Batch size*: 32
3. *Optimizer*: Adam
4. *Loss Function*: *Categorical Cross-Entropy*
5. *GPU* A100

Penelitian ini memakai tiga arsitektur: *MobileNetV2*, *VGG16*, dan *ResNet50*. Grafik menampilkan akurasi validasi (Val Accuracy) per *epoch*, sementara tabel di bawahnya merangkum *epoch* terbaik, akurasi terbaik, val loss terbaik, serta rata-rata waktu per *epoch* (*epoch\_seconds*). Tujuannya adalah memilih *model* terbaik dengan mempertimbangkan kualitas (akurasi) dan efisiensi waktu pelatihan.



Gambar 5. Grafik pemilihan model terbaik

Di bawah grafik: secara keseluruhan, *VGG16* dan *ResNet50* sama-sama tembus val accuracy = 1.000; *VGG16* memberikan val loss terbaik ( $\approx 0.0373$ ) namun dengan waktu/epoch paling lama ( $\approx 9:08/epoch$ ) sehingga total waktu ke performa terbaik  $\approx 4:48:51$ —cocok bila prioritas mutlak adalah akurasi/keluwasan kapasitas model dan waktu bukan kendala. *ResNet50* mencapai 1.000 paling cepat dalam jumlah epoch ( $\approx 13$  epoch), tetapi setiap epoch lebih berat ( $\approx 3:03/epoch$ ; time-to-best  $\approx 0:35:46$ ). *MobileNetV2* berada sedikit di bawah puncak (best val acc  $\approx 0.9855$ ), tetapi paling efisien waktu ( $\approx 0:57/epoch$ ; time-to-best  $\approx 0:14:25$ ) dan cenderung stabil mulai sekitar epoch 15—paling pas untuk kebutuhan produksi/real-time dan biaya komputasi rendah. Praktiknya: pilih *VGG16* untuk keseimbangan, *ResNet50* untuk akurasi maksimum, dan *MobilenetV2* untuk kecepatan/biaya.

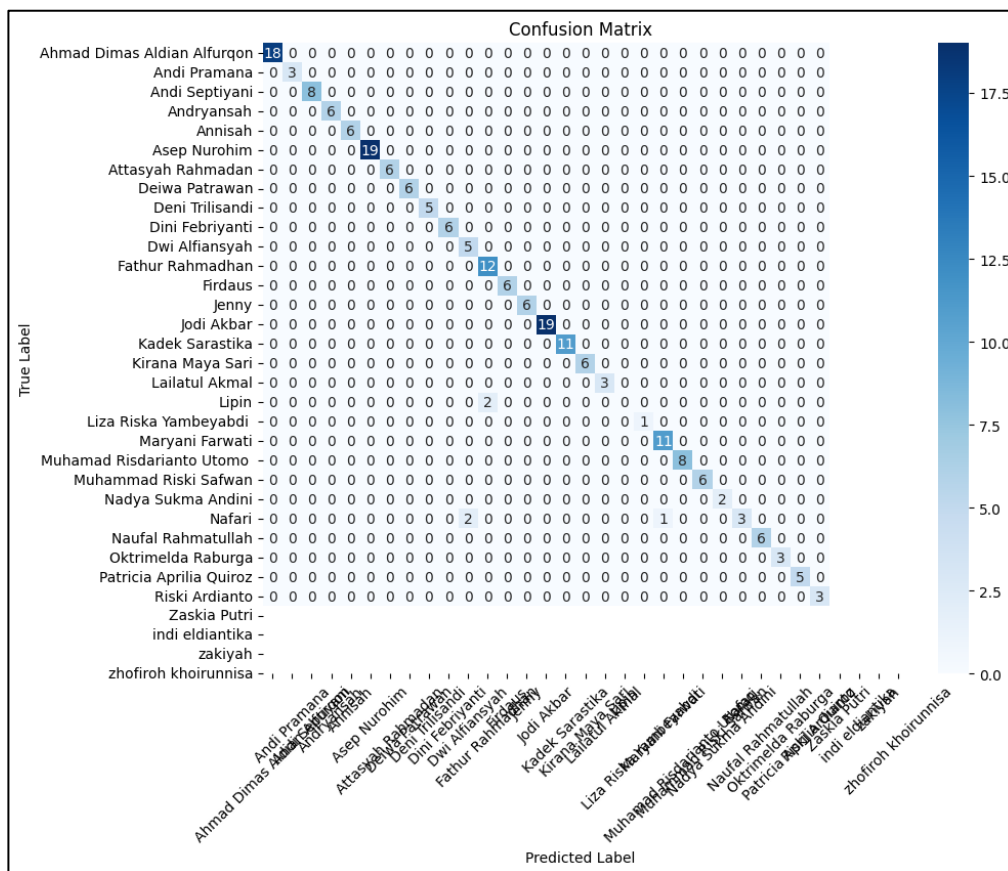
Tabel 1. Perbandingan Model

| model       | Best epoch | Best accuracy | Best val loss | Best val loss | Best Val accuracy | Time_epoch     |
|-------------|------------|---------------|---------------|---------------|-------------------|----------------|
| MobileNetV2 | 15         | 0.9700        | 0.2239        | 0.2129        | 0.9855            | <b>865 S</b>   |
| VGG16       | 32         | 0.9829        | 0.0593        | 0.0373        | 1.0000            | <b>17331 S</b> |
| ResNet50    | 13         | 0.9849        | 0.1232        | 0.0783        | 1.0000            | <b>2146 S</b>  |

*MobileNetV2* dipilih karena arsitektur ini memberikan kombinasi terbaik antara akurasi validasi yang tinggi dan efisiensi waktu pelatihan pada eksperimen yang dilakukan, sekaligus menghasilkan kurva validasi yang relatif stabil. Secara teoritis, keunggulan *MobileNetV2* terletak pada desain *konvolusi separable yang mendalam* yang mampu mengekstrak representasi fitur kaya dengan jumlah parameter yang

jauh lebih sedikit dibandingkan arsitektur konvensional, sehingga sangat sesuai untuk implementasi pada perangkat dengan keterbatasan *komputasi*. Dalam praktik penelitian, akurasi validasi maksimum merupakan indikator yang paling dekat dengan kemampuan generalisasi model pada data tak terlihat—tujuan utama dari proses pelatihan. Hal ini sejalan dengan evaluasi literatur klasifikasi yang tegangan intensitas akurasi (bersama metrik lain) sebagai dasar perbandingan model serta pembenaran keputusan *penerapan*, terutama ketika distribusi kelas relatif seimbang dan kebutuhan aplikasi menuntut prediksi yang tepat sebanyak mungkin.

### 3.4.1 Confusion Matrix



Gambar 6. Confusion Matrix

Gambar di atas adalah visualisasi *Confusion Matrix* yang menampilkan performa model pengenalan wajah dalam mengklasifikasikan 520 gambar testing ke dalam 33 kelas data mandiri yang berbeda.

Hasil *confusion matrix* ini menunjukkan bahwa model pengenalan wajah berbasis arsitektur *MobileNetV2* telah dilatih dan diuji dengan sukses, menghasilkan prediksi yang akurat dan tingkat kesalahan yang rendah pada data testing. Model ini siap untuk diterapkan dalam sistem pengenalan wajah yang lebih luas.

### 3.4.2 Clasification Report

|                             | precision | recall | f1-score | support |
|-----------------------------|-----------|--------|----------|---------|
| Ahmad Dimas Aldian Alfurqon | 1.00      | 1.00   | 1.00     | 18      |
| Andi Pramana                | 1.00      | 1.00   | 1.00     | 3       |
| Andi Septiyani              | 1.00      | 1.00   | 1.00     | 8       |
| Andryansah                  | 1.00      | 1.00   | 1.00     | 6       |
| Annisah                     | 1.00      | 1.00   | 1.00     | 6       |
| Asep Nurohim                | 1.00      | 1.00   | 1.00     | 19      |
| Attasyah Rahmadan           | 1.00      | 1.00   | 1.00     | 6       |
| Deiwa Patrawan              | 1.00      | 1.00   | 1.00     | 6       |
| Deni Trilisandi             | 1.00      | 1.00   | 1.00     | 5       |
| Dini Febriyanti             | 1.00      | 1.00   | 1.00     | 6       |
| Dwi Alfiansyah              | 0.71      | 1.00   | 0.83     | 5       |
| Fathur Rahmadhan            | 0.86      | 1.00   | 0.92     | 12      |
| Firdaus                     | 1.00      | 1.00   | 1.00     | 6       |
| Jenny                       | 1.00      | 1.00   | 1.00     | 6       |
| Jodi Akbar                  | 1.00      | 1.00   | 1.00     | 19      |
| Kadek Sarastika             | 1.00      | 1.00   | 1.00     | 11      |
| Kirana Maya Sari            | 1.00      | 1.00   | 1.00     | 6       |
| Lailatul Akmal              | 1.00      | 1.00   | 1.00     | 3       |
| Lipin                       | 0.00      | 0.00   | 0.00     | 2       |
| Liza Riska Yambeyabdi       | 1.00      | 1.00   | 1.00     | 1       |
| Maryani Farwati             | 0.92      | 1.00   | 0.96     | 11      |
| Muhamad Risdarianto Utomo   | 1.00      | 1.00   | 1.00     | 8       |
| Muhammad Riski Safwan       | 1.00      | 1.00   | 1.00     | 6       |
| Nadya Sukma Andini          | 1.00      | 1.00   | 1.00     | 2       |
| Nafari                      | 1.00      | 0.50   | 0.67     | 6       |
| Naufal Rahmatullah          | 1.00      | 1.00   | 1.00     | 6       |
| Oktrimeida Raburga          | 1.00      | 1.00   | 1.00     | 3       |
| Patricia Aprilia Quiroz     | 1.00      | 1.00   | 1.00     | 5       |
| Riski Ardianto              | 1.00      | 1.00   | 1.00     | 3       |
| accuracy                    |           |        | 0.98     | 204     |
| macro avg                   | 0.95      | 0.95   | 0.94     | 204     |
| weighted avg                | 0.97      | 0.98   | 0.97     | 204     |

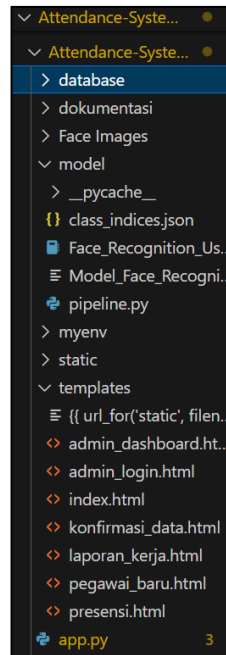
Gambar 7. Classification Report

Gambar di atas menunjukkan hasil classification report yang berisi evaluasi performa model pengenalan wajah berdasarkan *precision*, *recall*, *F1-score*, dan support untuk setiap kelas dataset yang diuji.

Hasil gambar ini menunjukkan bahwa model *MobileNetV2* berhasil mencapai performa yang sangat tinggi pada dataset pengenalan wajah. Setiap metrik evaluasi memberikan nilai sempurna 1.00, menandakan bahwa tidak ada kesalahan prediksi yang terjadi. Namun, perlu diperhatikan bahwa hasil ini bisa jadi dipengaruhi oleh data yang terstruktur dengan baik, dan kemungkinan besar model tidak menghadapi tantangan yang signifikan seperti noise atau variasi ekstrem pada gambar. Untuk validasi lebih lanjut, disarankan menggunakan dataset lain atau melakukan pengujian pada gambar di luar dataset yang dilatih.

### 3.5 Implementasi REST API dengan Flask

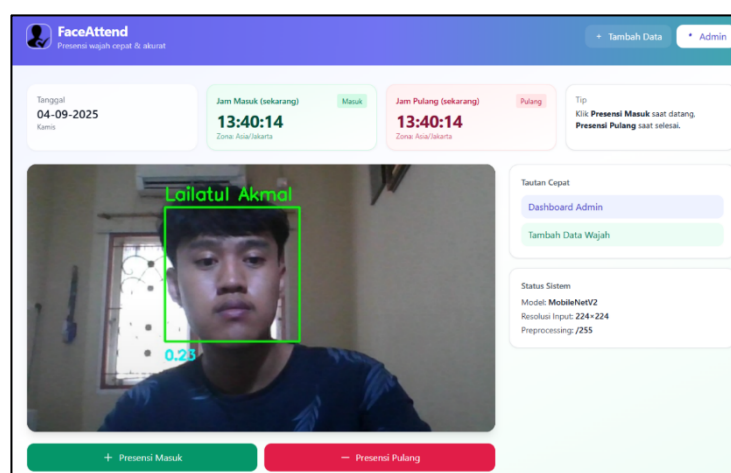
Aplikasi presensi ini dikembangkan menggunakan *Flask* sebagai *framework* utama berbasis *web*. *Flask* dipilih karena ringan, fleksibel, dan mendukung pengembangan *REST API* yang dapat diakses lintas *platform*. Dalam arsitektur sistem, *Flask* berperan sebagai *back-end* sekaligus penyedia layanan presensi berbasis pengenalan wajah dengan integrasi model *machine learning*.



Gambar 8. Struktur folder

### 3.6 Integrasi dengan Aplikasi Web

Aplikasi presensi berbasis web yang dikembangkan memiliki empat tampilan utama, yaitu halaman presensi, tambah data, dashboard admin, dan laporan presensi. Pada halaman presensi, sistem menampilkan deteksi wajah real-time dengan opsi presensi masuk dan pulang, serta informasi status pengguna. Halaman tambah data digunakan untuk perekaman citra wajah pegawai dengan pengambilan beberapa gambar dari berbagai sudut agar model lebih akurat. Dashboard admin menyediakan ringkasan statistik, daftar pegawai, dan akses cepat untuk pengelolaan data. Sementara itu, halaman laporan presensi memungkinkan admin memfilter data berdasarkan periode dan pegawai, serta mengekspor laporan dalam format CSV atau cetak.



Gambar 9. Pengujian aplikasi

### 3.7 Pengujian berbasis kamera web

Berikut adalah rangkuman pengujian fungsional (*black-box testing*) pada aplikasi *web* presensi berbasis pengenalan wajah yang dikembangkan menggunakan *Flask*. Pengujian ini berfokus pada perilaku sistem sebagaimana terlihat oleh pengguna, tanpa menilai kode internal maupun arsitektur program. Aspek yang diuji meliputi tampilan antarmuka, interaksi input dari kamera *web*, indikator status pada halaman presensi, konsistensi pengolahan data, serta penyajian laporan presensi.

Berdasarkan kriteria tersebut, hasil pengujian fungsional aplikasi *web* presensi dapat dirangkum dalam Tabel 2. berikut.

Tabel 2. Pengujian Aplikasi

| No | Tampilan Uji           | Tujuan Pengujian   | Langkah Uji   | Hasil  |
|----|------------------------|--|---|--|
| 1  | Halaman Utama Presensi | Memastikan sistem mampu mendeteksi wajah secara <i>real-time</i> dan mencatat presensi masuk/pulang. | <ol style="list-style-type: none"> <li>1) Buka halaman presensi melalui browser.</li> <li>2) Aktifkan kamera web.</li> <li>3) Arahkan wajah ke kamera.</li> <li>4) Klik tombol Presensi Masuk atau Presensi Pulang.</li> <li>5) Periksa apakah data tersimpan di database MySQL.</li> </ol> | Wajah berhasil terdeteksi, kotak deteksi muncul, status kehadiran tercatat sesuai tombol yang dipilih. |
| 2  | Halaman Tambah Data    | Menguji proses pendaftaran wajah baru pegawai dengan ID unik.  | <ol style="list-style-type: none"> <li>1) Buka halaman tambah data.</li> <li>2) Masukkan ID Pegawai.</li> <li>3) Ambil 10 citra wajah dengan variasi ekspresi dan sudut.</li> <li>4) Klik Ambil Gambar hingga progress bar penuh.</li> <li>5) Klik Reset untuk uji ulang.</li> </ol>        | Data wajah tersimpan dengan baik, sistem menerima 10 gambar, ID pegawai tercatat unik di database.     |

|   |                          |  |   |   |
|---|--------------------------|--|---|---|
| 3 | Halaman Admin Dashboard  | Memastikan admin dapat memantau statistik presensi dan mengelola data pegawai. | <ol style="list-style-type: none"> <li>1) Login sebagai admin.</li> <li>2) Buka Dashboard.</li> <li>3) Amati ringkasan statistik presensi (hari ini dan total).</li> <li>4) Klik menu Lihat Presensi pada salah satu pegawai.</li> <li>5) Uji tombol aksi cepat Tambah Data.</li> </ol> | Statistik tampil dengan benar, daftar pegawai dapat diakses, fitur aksi cepat berjalan tanpa error.       |
| 4 | Halaman Laporan Presensi | Menguji penyajian laporan presensi berdasarkan filter dan ekspor data.         | <ol style="list-style-type: none"> <li>1) Buka halaman laporan.</li> <li>2) Masukkan filter tanggal awal-akhir.</li> <li>3) Pilih nama pegawai tertentu.</li> <li>4) Klik Cari.</li> <li>5) Uji tombol Ekspor CSV dan Cetak.</li> </ol>   | Laporan tampil sesuai filter, tabel data dapat diekspor ke CSV, opsi cetak menghasilkan format yang rapi. |

#### 4. KESIMPULAN

Berdasarkan penelitian sistem pengenalan wajah menggunakan arsitektur MobileNetV2 dengan dataset yang diambil secara mandiri, dapat disimpulkan bahwa model mampu memberikan kinerja yang sangat baik dengan akurasi validasi mencapai 100%, serta didukung oleh penggunaan dropout dan augmentasi sederhana untuk mengurangi overfitting. Kualitas dan jumlah data terbukti sangat memengaruhi hasil, di mana kelas dengan jumlah data sedikit atau kualitas gambar rendah cenderung menghasilkan kesalahan prediksi. Augmentasi data yang digunakan mampu meningkatkan kemampuan generalisasi model, meskipun masih perlu dikembangkan dengan teknik yang lebih variatif seperti rotasi, flipping, dan penyesuaian pencahayaan. Hasil evaluasi menunjukkan performa yang sangat memuaskan dengan nilai precision, recall, dan f1-score sebesar 1.00 pada seluruh kelas, namun pengujian pada kondisi dunia nyata masih menunjukkan adanya kesalahan prediksi. Selain itu, variasi resolusi gambar yang tidak seragam juga memengaruhi hasil prediksi, terutama pada gambar dengan resolusi terlalu rendah atau terlalu tinggi. Ke depan, penelitian ini memiliki prospek untuk dikembangkan dengan penggunaan dataset yang lebih besar dan beragam, penerapan augmentasi yang lebih kompleks, serta pengujian dalam kondisi real-time. Integrasi dengan

sistem berbasis web atau IoT seperti absensi otomatis, serta eksplorasi arsitektur lain seperti EfficientNet atau FaceNet, juga menjadi peluang untuk meningkatkan performa dan ketahanan model.

## DAFTAR PUSTAKA

- [1] Cahyono, F., rachmandi, F. R., & Dea, W. (2020). *PENGENALAN WAJAH MENGGUNAKAN MODEL FACENET UNTUK PRESENSI PEGAWAI*.
- [2] Dewi, N., & Ismawan, F. (2021). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CNN UNTUK SISTEM PENGENALAN WAJAH. *Faktor Exacta*, 14(1), 34. <https://doi.org/10.30998/faktorexacta.v14i1.8989>
- [3] Firmansyah, A., Fauzul Itsnan, A., Apip, A., Tri Mulliya, R., & Rosyani, P. (2024). SISTEM ABSENSI MAHASISWA MENGGUNAKAN FACE RECOGNITION DENGAN ALGORITMA CNN. In *Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan* (Vol. 1, Issue 4). <https://jurnalmahasiswa.com/index.php/aidanspk>
- [4] Khatib Sulaiman, J., Gunawan Ramdhani, S., Itje Sela, E., & Teknologi Yogyakarta, U. (n.d.). Implementasi Face Recognition Untuk Sistem Presensi Universitas Menggunakan Convolutional Neural Network. *Indonesian Journal of Computer Science Attribution*, 12(6), 2023–4098.
- [5] Manihuruk, C., Fikry, M., Al, H., & Aidilof, K. (n.d.). *Face Recognition System For Student Identification Using Vgg16 Convolutional Neural Network*. <https://doi.org/10.29103/icomden.v2.xxxx>
- [6] Marvelino Wijaya, A., & Eric Samodra, J. (2023). *Wijaya, Sistem Presensi Pegawai Dengan Face Recognition Menggunakan Deep Learning CNN 163 Sistem Presensi Pegawai dengan Face Recognition Menggunakan Deep Learning CNN*.
- [7] Pratama, Y., Ginting, L. M., Nainggolan, E. H. L., & Rismanda, A. E. (2021). Face recognition for presence system by using residual networks-50 architecture. *International Journal of Electrical and Computer Engineering*, 11(6), 5488–5496. <https://doi.org/10.11591/ijece.v11i6.pp5488-5496>
- [8] Riziq sirfatullah Alfarizi, M., Zidan Al-farish, M., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. In *Karimah Tauhid* (Vol. 2, Issue 1).
- [9] Saputra, R. J., Saragih, Y., Stefani, A., Universitas, D., & Karawang Abstract, S. (n.d.). Pendeteksi Face Mask Menggunakan Model CNN (Convolutional Neural Network). *Jurnal Ilmiah Wahana Pendidikan*, Desember, 2023(24), 600–606. <https://doi.org/10.5281/zenodo.10435426>
- [10] Sri, M. B., Srihari Rao, K., Anvitha, T., Anusha, V., Kamal, N. R., & Jayadweep, T. (2024). *Facial Attendance System using Flask*. [www.irjet.net](http://www.irjet.net)
- [11] Kulsum, U., & Cherid, A. (2023). Penerapan convolutional neural network pada klasifikasi tanaman menggunakan ResNet50. *Jurnal Sistem Informasi dan Sistem Komputer*, 8(2), 8 hlm.
- [12] Ismunandar, D., Firdaus, M. R., & Alkhalifi, Y. (2024). Penerapan hyperparameter machine learning dalam prediksi gagal pinjam. *INTI Nusa Mandiri*, 19(1), 9 hlm

- [13] Wilyani, F., Arif, Q. N., & Aslimar, F. (2024). Pengenalan dasar pemrograman Python dengan Google Colaboratory. *Jurnal Pengabdian Pada Masyarakat Indonesia*, 3(1), 7 hlm.
- [14] Riyadi, A. S., Wardhani, I. P., & Widayati, S. (2021). Klasifikasi citra anjing dan kucing menggunakan metode convolutional neural network (CNN). *Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA)*, 5(1), 5 hlm.
- [15] Susim, T., & Darujati, C. (2021). Pengolahan citra untuk pengenalan wajah (face recognition) menggunakan OpenCV. *Jurnal Syntax Admiration*, 2(3), 12 hlm.