



## IMPLEMENTASI SISTEM PRESENSI MAHASISWA BERBASIS PENGENALAN WAJAH MENGGUNAKAN MODEL *FACENET* (*CNN BACKBONE*) DAN METODE *KNN*

Maliki<sup>1</sup>, Lintang Aisah Lestari<sup>2</sup>, Khairul Rizal<sup>3</sup>, Susliansyah<sup>4</sup>, Rahmat Hidayat<sup>5</sup>

<sup>1</sup>maliki71004@gmail.com, <sup>2</sup>lintangaisah069@gmail.com, <sup>3</sup>khairul.krl@bsi.ac.id,

<sup>4</sup>susliansyah.slx@bsi.ac.id, <sup>5</sup>rahmat.rhh@bsi.ac.id

<sup>1,2,3,4,5</sup>Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

### Abstrak

Kemajuan teknologi kecerdasan buatan dan pemrosesan citra mendorong pengembangan sistem presensi yang lebih efektif dan akurat pada lingkungan perguruan tinggi. Penelitian ini difokuskan untuk menyusun dan mengimplementasikan sistem presensi mahasiswa menggunakan pengenalan wajah menggunakan *FaceNet* (*CNN Backbone*) sebagai ekstraktor fitur, sedangkan metode *K-Nearest Neighbors* (*KNN*) untuk klasifikasi identitas. Penelitian dilakukan melalui delapan tahapan, yaitu *data collecting*, *exploratory data analysis*, *data preprocessing*, *model training*, *model evaluation*, *deployment*, *system testing*, dan *monitoring*. Dataset yang digunakan terdiri dari 325 citra wajah dari 13 kelas mahasiswa dengan variasi *pose*, ekspresi, dan kondisi pencahayaan. Pengujian sistem dilakukan pada beberapa skenario utama, termasuk variasi intensitas cahaya dan jarak wajah ke kamera. Hasil pengujian menunjukkan akurasi pengenalan wajah sebesar 97% pada dataset yang digunakan dengan performa sistem yang stabil. Hasil ini menunjukkan bahwa kombinasi *FaceNet* (*CNN Backbone*) dan *KNN* efektif untuk mendukung presensi mahasiswa otomatis di lingkungan perguruan tinggi.

**Kata kunci:** *FaceNet*, *KNN*, Presensi Mahasiswa, Pengenalan Wajah.

### Abstract

*Advances in artificial intelligence and image processing technology are driving the development of more effective and accurate attendance systems in higher education environments. This research focuses on developing and implementing a student attendance system using facial recognition using FaceNet (CNN backbone) as a feature extractor, while the K-Nearest Neighbors (KNN) method is used for identity classification. The research was conducted through eight stages, namely data collection, exploratory data analysis, data preprocessing, model training, model evaluation, deployment, system testing, and monitoring. The dataset used consists of 325 facial images from 13 classes of students with various poses, expressions, and lighting conditions. System testing was conducted in several key scenarios, including variations in light intensity and face-to-camera distance. The test results showed a facial recognition accuracy of 97% on the dataset used with stable system performance. These results indicate that the combination of FaceNet (CNN backbone) and KNN is effective in supporting automatic student attendance in higher education environments.*

**Keywords:** *FaceNet*, *KNN*, *Student Attendance*, *Facial Recognition*.

### 1. Pendahuluan

Presensi merupakan aktivitas yang dilakukan untuk memantau tingkat kehadiran individu pada instansi, sekolah atau perusahaan[1]. Di lingkungan perguruan tinggi, proses presensi mahasiswa masih banyak yang dilakukan secara konvensional, sehingga mudah terjadi kesalahan dalam pencatatan, manipulasi kehadiran, dan membutuhkan waktu yang relatif lama. Kondisi ini mendorong perlunya sistem presensi yang lebih efisien, akurat, dan aman dengan meminimalkan intervensi manusia.

Kecerdasan buatan atau *Artificial Intelligence* (AI) adalah program komputer yang dikembangkan untuk meniru kecerdasan manusia[2]. Sedangkan *Computer Vision* adalah sistem yang

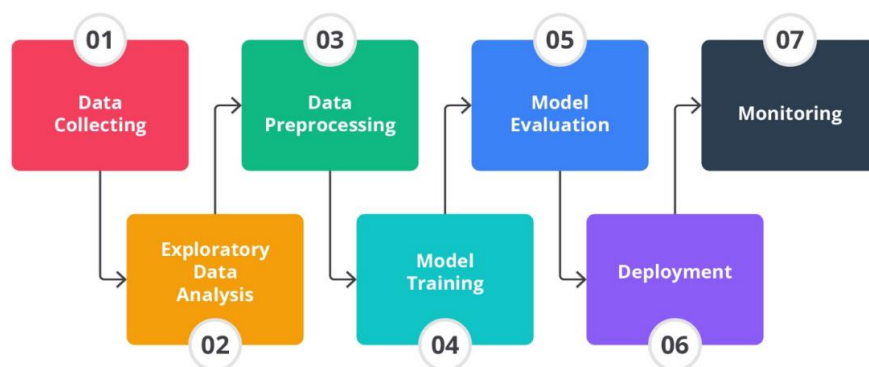
dapat melakukan pemahaman tingkat tinggi dengan mengajarkan mesin, dalam hal ini komputer dapat mengenali objek atau pendekatan yang mirip dengan sistem visual manusia di dunia nyata [3]. Perkembangan *AI* dan *computer vision* membuka peluang penerapan pengenalan wajah sebagai metode otomatis untuk identifikasi individu. *FaceNet (CNN backbone)* dapat mengekstraksi fitur wajah dalam bentuk *embedding* berdimensi rendah namun representatif, sehingga efektif dalam membedakan identitas. Proses klasifikasi *embedding* tersebut dapat dilakukan menggunakan metode *K-Nearest Neighbors (KNN)* yang bersifat sederhana dan efisien.

Beberapa penelitian sebelumnya telah menerapkan *face recognition* untuk sistem presensi. Penelitian oleh [4] berjudul “Pengolahan Citra untuk Pengenalan Wajah (*Face Recognition*) Menggunakan *OpenCV*”, melaporkan akurasi sekitar 85%, namun performanya masih dipengaruhi oleh variasi pencahayaan dan posisi wajah. Penelitian lain oleh [5] berjudul “Perancangan Sistem Absen Berbasis *Face Recognition*” juga memperoleh akurasi serupa yaitu 85% dengan pendekatan *deep learning*, tetapi mengalami kendala kestabilan deteksi wajah pada kondisi lingkungan nyata. Keterbatasan tersebut menunjukkan bahwa sistem presensi berbasis pengenalan wajah masih memerlukan perbaikan dari sisi akurasi dan kestabilan, khususnya melalui pemilihan model ekstraksi fitur yang lebih kuat serta metode klasifikasi yang sesuai.

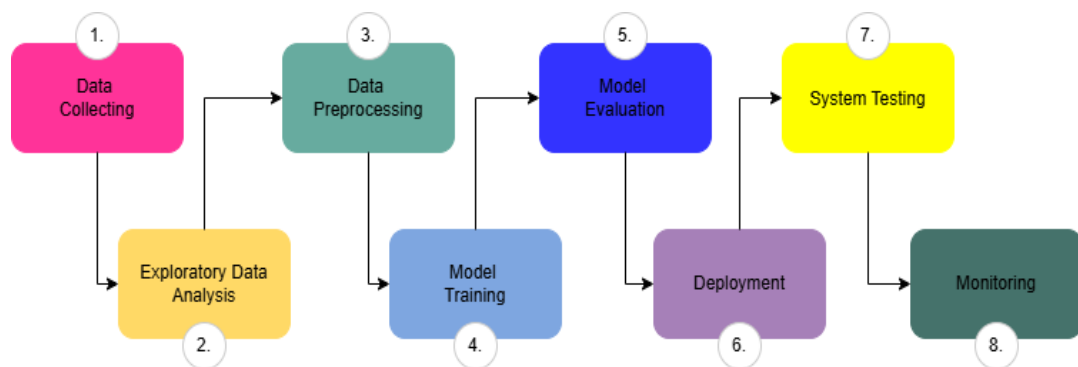
Dalam penelitian ini model *FaceNet* memanfaatkan arsitektur *Convolutional Neural Network (CNN)* sebagai *backbone* untuk mengambil fitur wajah secara akurat dengan kebutuhan komputasi yang efisien. *Embedding* wajah yang dihasilkan selanjutnya diklasifikasikan menggunakan metode *K-Nearest Neighbor (KNN)* yang digunakan karena kesederhanaannya serta kinerja metode tersebut dalam mengukur kemiripan antar data[6]. Sistem yang dikembangkan dirancang untuk bekerja secara otomatis dan *real-time* dalam proses presensi mahasiswa menggunakan pengenalan wajah. Tujuan penelitian ini adalah untuk menerapkan dan melakukan evaluasi sistem presensi mahasiswa menggunakan pengenalan wajah, dengan menggunakan kombinasi *FaceNet (CNN Backbone)* dan *KNN* guna meningkatkan efektivitas, akurasi, dan keamanan proses presensi di lingkungan perguruan tinggi.

## 2. Metode

Pada penelitian ini proses pengembangan sistem mengikuti metode *Machine Learning Pipeline* yang terdiri dari tujuh langkah utama. Sebuah data *pipeline* adalah rangkaian proses komputasi otomatis yang memindahkan dan memproses data dari sumber ke sistem tujuan, sehingga data siap untuk dianalisis[7]. *Pipeline* ini digunakan untuk menjamin bahwa sistem yang dibuat memiliki tingkat akurasi yang tinggi, stabil, serta dapat digunakan secara *real-time* dalam proses presensi mahasiswa. Metode yang digunakan mengikuti metode *Machine Learning Pipeline* yang bersumber dari *website*[8] dengan penambahan tahap *System Testing* setelah *Development* untuk menyesuaikan kebutuhan pengujian sistem presensi *real-time* sebelum tahap *Monitoring*. Adapun tahapan yang dilakukan dijelaskan sebagai berikut.



Gambar 1. Tahapan Metode *Machine Learning Pipeline*
















Gambar 2. Tahapan Penelitian

## 2.1 Data Collecting

Tahap pertama adalah *Data Collecting*, *Data collecting* adalah proses pengumpulan, pengukuran, dan analisis berbagai jenis data dengan menggunakan beberapa metode[9]. Proses ini penting untuk menjamin konsistensi dataset dan mendukung performa optimal model *FaceNet* dalam mengekstraksi fitur wajah mahasiswa.

Tabel 1. Citra Wajah Mahasiswa

No	Nama Citra	Citra Wajah
1.	Bintang	
2.	Dafa	
3.	Dyon	
4.	Gadis	
5.	Lintang	
6.	Maliki	
7.	Maria	

8.	Radit	
9.	Rifal	
10.	Ryan	
11.	Sherly	
12.	Syafiq	
13.	Wahyu	

Dataset penelitian ini terdiri dari 13 kelas identitas mahasiswa dengan total 325 citra wajah yang diperoleh melalui pemotretan langsung menggunakan kamera *smartphone* pada lingkungan *indoor*. Setiap kelas direkam sebanyak 25 citra dengan variasi *pose*, ekspresi wajah, jarak pengambilan gambar, dan intensitas pencahayaan, serta disimpan dalam format *JPEG* dengan resolusi 280×280 piksel. Variasi alami pada dataset dinilai cukup untuk mendukung proses ekstraksi fitur oleh model *FaceNet*, sehingga augmentasi data sintesis tidak diterapkan guna mempertahankan representasi kondisi nyata. Untuk keperluan dokumentasi, Tabel 1 hanya menampilkan citra *frontal* sebagai representasi tiap kelas, sementara seluruh variasi citra tetap digunakan dalam tahapan analisis dan pemrosesan data.

## 2.2 Exploratory Data Analysis

Tahap kedua adalah *Exploratory Data Analysis (EDA)*, *EDA* adalah sebuah tahapan dalam data *science* untuk mengenal dan memahami karakteristik dataset[10]. *EDA* dilakukan untuk mengidentifikasi pola, variasi, serta potensi masalah yang dapat memengaruhi performa model *FaceNet* dalam menghasilkan *embedding* wajah.

Pada penelitian ini, *EDA* difokuskan pada beberapa aspek berikut:

1. Distribusi jumlah citra per mahasiswa  
Analisis ini memastikan setiap mahasiswa memiliki jumlah data yang relatif seimbang.
2. Variasi kondisi pencahayaan dan pose wajah  
Dataset diperiksa untuk melihat keberagaman kondisi seperti intensitas cahaya, arah cahaya, latar belakang, serta sedikit variasi pose (miring kanan, kiri, atau sedikit mendongak).
3. Kualitas citra dan tingkat noise.  
Citra dianalisis untuk mendeteksi potensi noise seperti buram, bayangan kuat, atau bagian wajah yang tidak lengkap. Citra yang memiliki kualitas rendah dicatat agar diperbaiki pada tahap *preprocessing*.

4. Deteksi awal adanya *outlier*

*Outlier* berupa citra yang bukan wajah atau citra wajah yang tidak sesuai standar diidentifikasi untuk dihapus agar tidak memengaruhi hasil *embedding*.

Melalui proses *EDA* ini, kondisi dataset dapat dipahami dengan lebih baik sehingga tahap *preprocessing* dan pelatihan model dapat dirancang secara tepat. *EDA* berperan penting dalam memastikan bahwa model *FaceNet backbone CNN* bekerja pada data yang bersih, konsisten, dan representatif.

### 2.3 Data Preprocessing

Tahap ketiga setelah melakukan *Exploratory Data Analysis (EDA)* ialah proses *data preprocessing*. *Data preprocessing* atau pra-pemrosesan data, merupakan rangkaian langkah atau tahapan yang diterapkan pada data mentah sebelum data tersebut dipakai dalam analisis lebih lanjut atau pengembangan model[11]. Tahapan ini dilakukan untuk memastikan bahwa citra wajah berada dalam kondisi yang optimal sebelum memasuki proses ekstraksi fitur oleh model *FastNet-CNN*.

Pada penelitian ini proses *preprocessing* dilakukan *Cleaning*, *Cleaning* dilakukan untuk memastikan hanya citra yang valid dan sesuai standar yang dilibatkan dalam proses pelatihan model. Pada tahap ini, citra dengan kualitas rendah seperti gambar buram, pencahayaan terlalu gelap atau terlalu terang, wajah tidak menghadap kamera, serta *frame* yang tidak berisi wajah, akan dihapus.

### 2.4 Model Training

Tahap keempat adalah *Model Training*. Pada tahap *Model Training*, proses pelatihan dilakukan dengan memanfaatkan model *FaceNet* sebagai *feature extractor* dan algoritma KNN sebagai metode klasifikasi akhir. *FaceNet* dimanfaatkan untuk mengekstraksi fitur wajah dengan mengonversi setiap citra menjadi vektor *embedding* berdimensi 128 yang merepresentasikan karakteristik unik tiap mahasiswa.

Dataset yang telah melalui proses *preprocessing* kemudian dipisahkan menjadi dua bagian, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*) dengan perbandingan 80% untuk pelatihan dan 20% untuk pengujian. Pembagian ini dilakukan untuk memastikan model memiliki data yang memadai dalam proses pembelajaran model dan evaluasi kinerja secara objektif. Selama proses pelatihan, setiap citra pada *training set* diekstraksi menggunakan *FaceNet* sehingga menghasilkan *embedding*. *Embedding* tersebut kemudian digunakan untuk melatih algoritma KNN agar mampu mengelompokkan dan mengenali identitas wajah berdasarkan kemiripan jarak vektornya.

### 2.5 Model Evaluation

Tahap kelima adalah *Model Evaluation*, tahap *Model Evaluation* dilakukan untuk mengukur sejauh mana model *FaceNet-CNN Backbone* yang kemudian dikombinasikan dengan metode *K-Nearest Neighbor (KNN)* mampu mengenali wajah mahasiswa secara akurat. Evaluasi tidak hanya berfokus pada satu metrik, karena satu metrik saja sering menyesatkan (*misleading*). Oleh karena itu digunakan empat metrik utama yaitu: *accuracy*, *precision*, *recall*, dan *F1-score*.

1. *Accuracy*

*Accuracy* menghitung proporsi prediksi yang tepat terhadap seluruh data uji.

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

2. *Precision*

*Precision* mengukur tingkat ketepatan model dalam memprediksi kelas positif.

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

### 3. Recall

*Recall* mengukur tingkat keberhasilan sistem dalam mengenali wajah.

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

### 4. F1-score

*F1-score* atau *F-Measure* merupakan rata-rata harmonik dari *precision* dan *recall*.

$$\text{F1 - score } 2x = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Keterangan:

- *True Positives (TP)* : Wajah mahasiswa terdaftar, berhasil dikenali.
- *True Negatives (TN)* : Wajah mahasiswa tidak terdaftar, berhasil tidak dikenali.
- *False Positives (FP)* : Wajah mahasiswa tidak terdaftar, berhasil dikenali sebagai mahasiswa.
- *False Negatives (FN)*: Wajah mahasiswa terdaftar, tidak berhasil dikenali.

## 2.6 Deployment

Tahap keenam yaitu *Deployment*, tahap *Deployment* merupakan proses penerapan suatu aplikasi, sistem, solusi, atau teknologi ke dalam lingkungan operasional yang sesungguhnya setelah melalui tahap pengembangan (*development*)[12]. Pada tahap ini, model *FaceNet* dengan *backbone CNN* yang menghasilkan *embedding* wajah diintegrasikan dengan algoritma *K-Nearest Neighbor (KNN)* sebagai pengklasifikasi guna mengenali identitas mahasiswa secara *real-time*.

Model hasil pelatihan diimplementasikan pada *web* berbasis bahasa pemrograman *Python* menggunakan *framework Flask*. Selanjutnya, modul pemrosesan citra diintegrasikan menggunakan *library OpenCV* agar sistem dapat menangkap *frame* menggunakan kamera bawaan laptop, melakukan *face detection*, mengekstraksi *embedding* menggunakan *FaceNet*, lalu membandingkannya dengan *embedding* pada basis data menggunakan *KNN*.

Sistem presensi juga dihubungkan dengan basis data untuk menyimpan informasi identitas mahasiswa, *embedding* wajah, serta hasil kehadiran. Saat mahasiswa berada di depan kamera, sistem secara otomatis mendeteksi wajah, melakukan pencocokan identitas, lalu mencatat presensi ke dalam basis data secara *real-time*. Dengan demikian, *proses deployment* memastikan model tidak hanya berjalan secara teoritis, tetapi dapat berfungsi sebagai sistem presensi yang lengkap, interaktif, dan siap digunakan di lingkungan kampus.

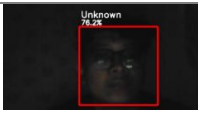


## 2.7 System Testing

Tahap ketujuh yaitu *System Testing*, *System Testing* adalah tahap pengujian perangkat lunak yang diterapkan pada sistem yang sudah sepenuhnya terintegrasi untuk menjamin bahwa sistem telah memenuhi seluruh persyaratan yang ditetapkan[13]. Pengujian ini tidak lagi berfokus pada performa model, tetapi pada kinerja keseluruhan sistem setelah diintegrasikan dengan kamera laptop dan antarmuka.

### 2.7.1 Pengujian Berdasarkan Intensitas Cahaya

Pada tahap ini, sistem diuji untuk menilai sejauh mana kemampuan model dalam mengenali wajah dapat bekerja secara konsisten pada kondisi pencahayaan yang berbeda.

Tabel 2. Pengujian Pada Intensitas Cahaya

No	Gambar	Skenario	Hasil	Keterangan
1.		Cahaya Redup	Tidak Terdeteksi	Citra wajah tidak teridentifikasi pada intensitas cahaya redup.
2.		Cahaya Normal	Terdeteksi	Citra wajah teridentifikasi pada intensitas cahaya normal.
3.		Cahaya Terang	Terdeteksi	Citra wajah teridentifikasi pada intensitas cahaya terang.

Pada Tabel 2 menunjukkan uji coba cahaya, di mana pada intensitas 1 citra wajah tidak terdeteksi, sedangkan intensitas 2 dan 3 terdeteksi. Hal ini menunjukkan bahwa cahaya dengan tingkat rendah memengaruhi proses pengenalan wajah.

### 2.7.2 Pengujian Berdasarkan Jarak

Pada tahap ini, sistem diuji untuk menilai seberapa konsisten kinerja model pengenalan wajah ketika ada variasi jarak antara wajah dan kamera.

Tabel 3. Pengujian Berdasarkan Jarak

No	Gambar	Skenario	Hasil	Keterangan
1.		Jarak 20cm	Terdeteksi	Citra wajah teridentifikasi pada jarak 20cm.
2.		Jarak 40cm	Terdeteksi	Citra wajah teridentifikasi pada jarak 40cm.
3.		Jarak 60cm	Terdeteksi	Citra wajah teridentifikasi pada jarak 60cm.
4.		Jarak 80cm	Terdeteksi	Citra wajah teridentifikasi pada jarak 80cm.

Pada Tabel 3 menunjukkan hasil uji coba berdasarkan jarak, di mana dilakukan empat percobaan pada jarak 20 cm, 40 cm, 60 cm, dan 80 cm. Hasil pengujian menunjukkan bahwa citra wajah pada semua jarak tersebut berhasil terdeteksi. Temuan ini menunjukkan bahwa kinerja sistem mampu melakukan proses pengenalan wajah dengan stabil pada rentang jarak 20 – 80 cm.

### 2.8 Monitoring

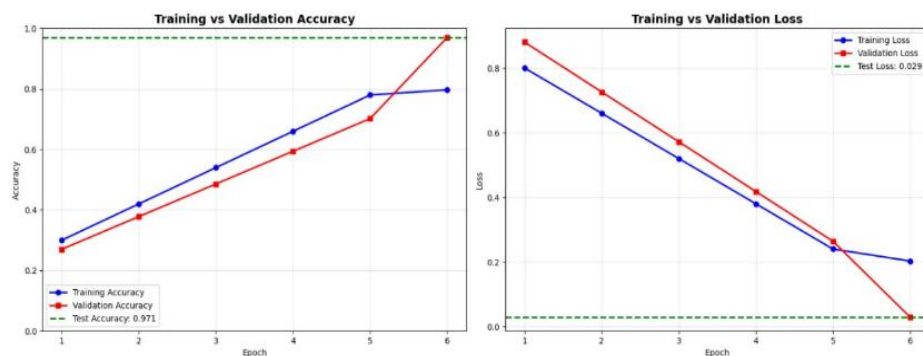
Tahap kedelapan yaitu *Monitoring*, tahap *Monitoring* merupakan pemantauan perkembangan tahapan-tahapan yang telah dilakukan[14]. Pemantauan dilakukan secara berkala untuk memastikan proses deteksi serta pengenalan wajah tetap stabil dan konsisten. Fokus *monitoring* mencakup akurasi pengenalan wajah, waktu *respons* sistem, dan tingkat kegagalan deteksi pada berbagai kondisi

pencahayaan maupun variasi posisi wajah pengguna. Selama proses penggunaan, *log* sistem diperiksa untuk mengidentifikasi kemungkinan *error*, misalnya kegagalan membaca *frame* kamera, ketidakcocokan *embedding*, atau performa model *FaceNet* dan *KNN* yang menurun. Jika ditemukan ketidaksesuaian output atau penurunan akurasi, maka dilakukan evaluasi ulang terhadap dataset, termasuk penambahan citra baru mahasiswa, pembaruan *embedding*, atau penyesuaian parameter pada proses klasifikasi. Monitoring ini memastikan sistem tetap andal dalam jangka panjang, menjaga kestabilan proses presensi otomatis, serta menjamin bahwa deteksi wajah tetap berfungsi optimal meskipun terdapat perubahan kondisi lingkungan atau karakteristik pengguna.

### 3. Hasil dan Pembahasan

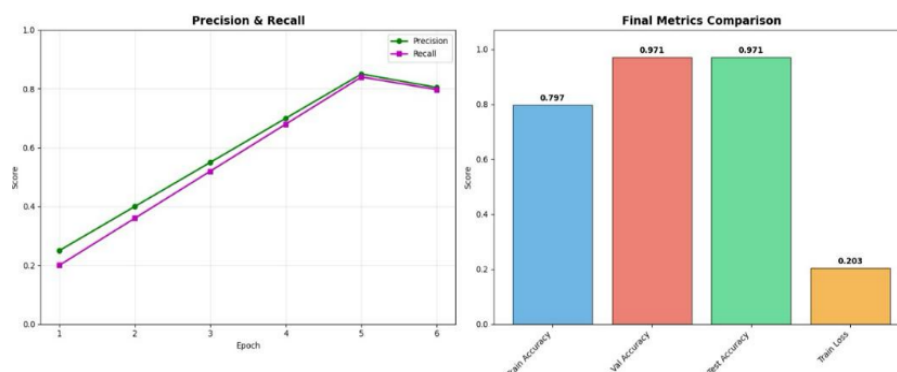
#### 3.1 Hasil Pelatihan Model

Tahap pelatihan model diimplementasikan dengan memanfaatkan dataset berjumlah 325 citra yang terdiri dari 13 kelas. Setiap kelas berisi sekitar 25 citra wajah, seluruh data diperoleh secara manual melalui pemotretan langsung menggunakan kamera *smartphone* yang terintegrasi dengan sistem. Variasi *pose* (*frontal*, miring ke kanan, miring ke kiri), perubahan ekspresi wajah, kondisi mata tertutup, perbedaan jarak pemotretan, serta variasi intensitas pencahayaan sengaja dilakukan tanpa augmentasi tambahan untuk memberikan keragaman data alami. Pendekatan ini bertujuan untuk menilai kemampuan model dalam melakukan pengenalan wajah mahasiswa secara konsisten meskipun terdapat perbedaan ekspresi dan kondisi pengambilan gambar pada situasi nyata. Berikut hasil pelatihan model.



Gambar 3. Training vs Validation Accuracy dan Training vs Validation Loss

Gambar 3 menunjukkan perbandingan *training vs validation accuracy* serta *training vs validation loss* selama proses pengembangan sistem presensi berbasis pengenalan wajah. Pada *training vs validation accuracy*, dengan nilai *training accuracy* meningkat secara konsisten pada setiap *epoch*, dengan akurasi pelatihan naik dari sekitar 30% hingga lebih dari 80%, sementara *validation accuracy* mencapai nilai mendekati 100% pada *epoch* terakhir, serta didukung oleh *test accuracy* sekitar 0,97. Pola peningkatan yang sejalan antara kedua kurva menunjukkan bahwa sistem mampu mempelajari fitur wajah secara efektif tanpa indikasi *overfitting* yang signifikan. Sementara itu *training vs validation loss*, nilai *training loss* menurun dari sekitar 0,8 menjadi 0,2 dan *validation loss* mendekati nol, dengan *test loss* yang sangat rendah yaitu sekitar 0,029, menandakan bahwa model telah mencapai konvergensi yang baik dan mampu meminimalkan kesalahan prediksi. Dalam konteks sistem presensi, kondisi ini penting untuk memastikan identifikasi mahasiswa yang stabil pada data baru.



Gambar 4. Precision & Recall dan Final Metrics Comparison

Gambar 4 menampilkan *precision & recall* serta perbandingan metrik akhir (*final metrics comparison*) dari sistem yang dikembangkan. Nilai *precision & recall* meningkat secara stabil hingga mencapai kisaran 0,82–0,85 pada *epoch* terakhir, yang menunjukkan bahwa sistem tidak hanya akurat dalam menetapkan identitas mahasiswa (*precision*), tetapi juga mampu mengenali sebagian besar wajah mahasiswa yang valid (*recall*). Pada *final metrics comparison*, *training accuracy* mencapai 0,797, sedangkan *accuracy validation* dan *test accuracy* memiliki nilai yang sama, yaitu 0,971, dengan *training loss* sebesar 0,203. Konsistensi performa antara data validasi dan pengujian ini mengindikasikan bahwa sistem presensi berbasis *FaceNet* dan *KNN* memiliki stabilitas dan kemampuan generalisasi yang baik untuk digunakan dalam lingkungan nyata.

### 3.2 Perhitungan Manual K-Nearest Neighbor (KNN)

Proses klasifikasi identitas mahasiswa dilakukan memanfaatkan metode *KNN* yang didasarkan pada nilai *embedding* yang dihasilkan oleh model *FaceNet* dengan *backbone CNN*. Setiap citra wajah yang masuk diproses oleh *FaceNet* dan diubah menjadi vektor berukuran 128 dimensi yang merepresentasikan ciri khas setiap wajah mahasiswa. Untuk melakukan identifikasi, *embedding* wajah uji dibandingkan dengan *embedding* wajah yang telah tersimpan di basis data.

Pada penelitian ini, contoh perhitungan manual dijelaskan menggunakan tiga data mahasiswa, yaitu Bintang, Ryan, dan Sherly. Dari setiap mahasiswa, peneliti mengambil satu *embedding* sebagai data latih dan satu *embedding* lain sebagai data uji untuk menunjukkan cara kerja perhitungan *KNN* secara manual. Mengingat bahwa setiap *embedding FaceNet* terdiri dari 128 vektor, maka untuk penyederhanaan proses perhitungan manual hanya digunakan 10 nilai pertama dari masing-masing vektor *embedding*. Metode *KNN* menggunakan jarak *Euclidean* sebagai ukuran kemiripan antar *embedding*.

Rumus jarak *Euclidean* didefinisikan sebagai berikut:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5)$$

Keterangan:

- d: *distance* (jarak)
- $x_i$ : nilai *embedding* data latih
- $y_i$ : nilai *embedding* data uji
- n: jumlah dimensi

Untuk perhitungan *K-Nearest Neighbor (KNN)* secara manual yaitu sebagai berikut:

1. Bintang

- *Embedding* Latih (E1):  
[0,1469, -0,0748, 0,0188, 0,0019, -0,0451, 0,0684, -0,0983, 0,0467, 0,0591, 0,0996]
- *Embedding* Uji (E2):  
[0,0391, -0,0483, 0,1081, 0,0473, -0,0327, 0,0332, 0,0406, 0,0923, 0,0483, 0,1245]

Tabel 4. Perhitungan Jarak Setiap Dimensi Bintang

Dim	$x_i$	$y_i$	$(x_i - y_i)$	$(x_i - y_i)^2$
1.	0,1469	0,0391	0,1078	0,011621
2.	-0,0748	-0,0483	-0,0265	0,000702
3.	0,0188	0,1081	-0,0893	0,007974
4.	0,0019	0,0473	-0,0454	0,002061
5.	-0,0451	-0,0327	-0,0124	0,000154
6.	0,0684	0,0332	0,0352	0,001239
7.	-0,0983	0,0406	-0,1389	0,019293
8.	0,0467	0,0923	-0,0456	0,002079
9.	0,0591	0,0483	0,0108	0,000117
10.	0,0996	0,1245	-0,0249	0,00062
$\Sigma$				<b>= 0,0458608</b>
$d = \sqrt{0,0458608}$				<b>= 0,2141513</b>

2. Ryan

- *Embedding* Latih (E1):  
[0.0762, 0.0950, -0.0668, -0.0385, 0.0518, -0.1813, -0.0172, -0.0692, -0.0671, 0.0015]
- *Embedding* Uji (E2):  
[0,0391, -0,0483, 0,1081, 0,0473, -0,0327, 0,0332, 0,0406, 0,0923, 0,0483, 0,1245]

Tabel 5. Perhitungan Jarak Setiap Dimensi Ryan

Dim	$x_i$	$y_i$	$(x_i - y_i)$	$(x_i - y_i)^2$
1.	0,0762	0,0391	0,0371	0,001376
2.	0,095	-0,0483	0,1433	0,020535
3.	-0,0668	0,1081	-0,1749	0,03059
4.	-0,0385	0,0473	-0,0858	0,007362
5.	0,0518	-0,0327	0,0845	0,00714
6.	-0,1813	0,0332	-0,2145	0,04601
7.	-0,0172	0,0406	-0,0578	0,003341
8.	-0,0692	0,0923	-0,1615	0,026082
9.	-0,0671	0,0483	-0,1154	0,013317
10.	0,0015	0,1245	-0,123	0,015129
$\Sigma$				<b>= 0,1708827</b>
$d = \sqrt{0,1708827}$				<b>= 0,4133796</b>

3. Sherly

- *Embedding* Latih (E1):  
[0.0257, 0.1498, -0.1108, -0.0318, 0.0153, -0.0936, -0.0581, -0.1362, 0.0364, 0.0110]
- *Embedding* Uji (E2):  
[0,0391, -0,0483, 0,1081, 0,0473, -0,0327, 0,0332, 0,0406, 0,0923, 0,0483, 0,1245]

Tabel 6. Perhitungan Jarak Setiap Dimensi Sherly

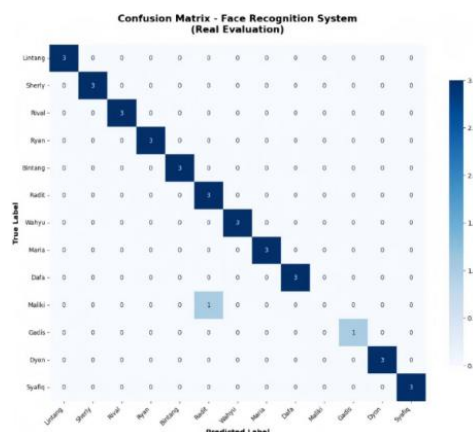
Dim	$x_i$	$y_i$	$(x_i - y_i)$	$(x_i - y_i)^2$
1.	0,0257	0,0391	-0,0134	0,00018
2.	0,1498	-0,0483	0,1981	0,039244
3.	-0,1108	0,1081	-0,2189	0,047917
4.	-0,0318	0,0473	-0,0791	0,006257
5.	0,0153	-0,0327	0,048	0,002304
6.	-0,0936	0,0332	-0,1268	0,016078
7.	-0,0581	0,0406	-0,0987	0,009742
8.	-0,1362	0,0923	-0,2285	0,052212
9.	0,0364	0,0483	-0,0119	0,000142
10.	0,011	0,1245	-0,1135	0,012882
$\Sigma$				<b>= 0,1869572</b>
$d = \sqrt{0,1869572}$				<b>= 0,4323855</b>

➤ Jadi  $K=1$ =Bintang, karena jarak *euclidean* milik Bintang yang paling kecil dibandingkan data latih lain.

1. Jarak ke Bintang = 0,2141513
2. Jarak ke Ryan = 0,4133796
3. Jarak ke Sherly = 0,4323855

### 3.3 Evaluasi Kinerja Model

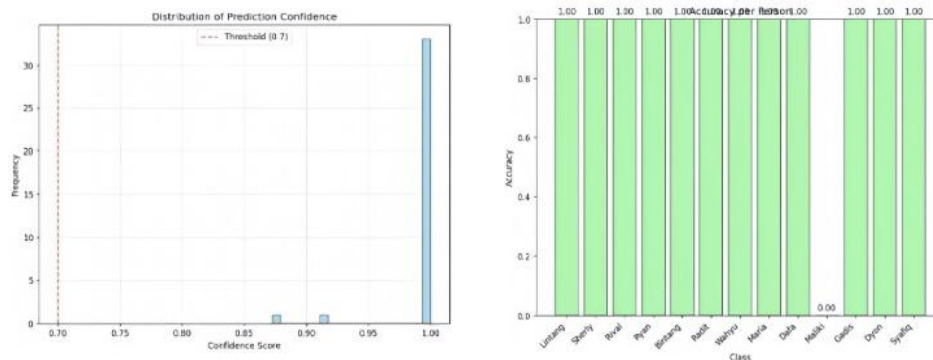
Evaluasi kinerja dilakukan untuk mengukur efektivitas mekanisme presensi mahasiswa dengan pendekatan pengenalan wajah yang dibangun memanfaatkan *FaceNet* pada tahap ekstraktor fitur dan metode *K-Nearest Neighbors* (KNN) untuk algoritma klasifikasi identitas. Evaluasi mencakup pengukuran akurasi prediksi, kesalahan klasifikasi, tingkat kepercayaan (*confidence*) model, serta performa pengenalan pada masing-masing identitas mahasiswa. Berbagai visualisasi digunakan, antara lain *confusion matrix*, *distribution of prediction confidence*, *accuracy per person*, *prediction error analysis*, *accuracy by confidence range* dan *classification report* guna memberikan gambaran menyeluruh terhadap kinerja sistem presensi yang dikembangkan.



Gambar 5. *Confusion Matrix*

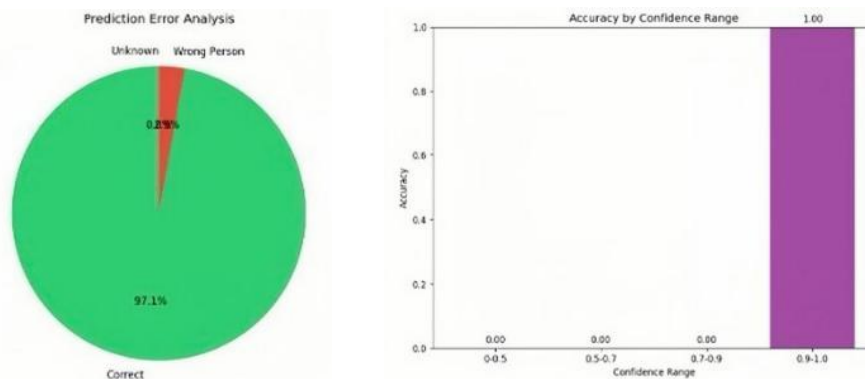
Gambar 5 menampilkan *confusion matrix* yang digunakan untuk mengevaluasi kesesuaian antara hasil prediksi model dan label sebenarnya pada seluruh kelas. Mayoritas nilai prediksi berada

pada diagonal *matriks*, yang menunjukkan tingkat klasifikasi yang tinggi. Sebagian besar kelas seperti Lintang, Shery, Rival, Ryan, Bintang, Radit, Wahyu, Maria, Dafa, Dyon, dan Syafiq mencapai tingkat pengenalan sempurna. Kesalahan klasifikasi hanya terjadi secara terbatas pada kelas Maliki dan Gadis, masing-masing pada satu sampel. Dalam konteks sistem presensi, kondisi ini menunjukkan bahwa kemungkinan kesalahan pencatatan kehadiran akibat salah identifikasi sangat kecil.



Gambar 6. *Distribution of Prediction Confidence dan Accuracy per Person*

Gambar 6 menyajikan *distribution of prediction confidence* dan *accuracy per person*. *Distribution of prediction confidence* menunjukkan bahwa sebagian besar prediksi berada pada rentang nilai tinggi, mendekati 1,0 dan secara konsisten berada di atas *threshold* identifikasi sebesar 0,7. Hal ini mengindikasikan bahwa sistem mampu memberikan keputusan identitas dengan tingkat keyakinan yang tinggi, sehingga mengurangi risiko presensi palsu atau tidak valid. *Accuracy per person* memperlihatkan bahwa hampir seluruh identitas mahasiswa mencapai akurasi 100%, menandakan konsistensi sistem dalam mengenali wajah mahasiswa yang sama pada berbagai kondisi pengambilan citra.



Gambar 7. *Prediction Error Analysis dan Accuracy By Confidence Range*

Gambar 7 menampilkan *prediction error analysis* dan *accuracy by confidence range*. *Prediction error analysis* mengelompokkan hasil prediksi ke dalam kategori *Correct*, *Wrong Person*, dan *Unknown*. Hasil menunjukkan bahwa sebesar 97,1% prediksi berada pada kategori *Correct*, yang berarti sistem berhasil mengenali identitas mahasiswa dengan tingkat keberhasilan yang sangat tinggi. Persentase kesalahan *Wrong Person* relatif kecil, sementara prediksi *Unknown* hampir tidak ditemukan, yang menunjukkan stabilitas sistem dalam proses identifikasi. *Accuracy by confidence range* memperlihatkan bahwa seluruh prediksi dengan *confidence* pada rentang 0,9–1,0 memiliki akurasi 100%, menegaskan bahwa nilai *confidence* yang dihasilkan model sejalan dengan ketepatan identifikasi dalam sistem presensi.

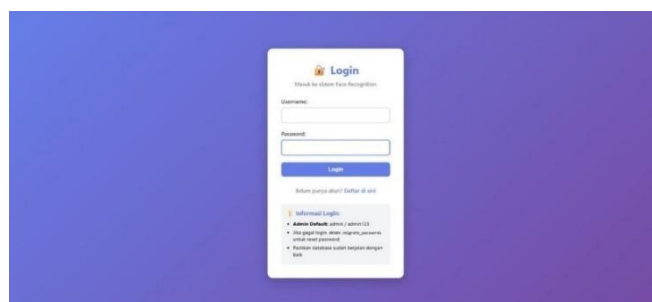
Class	Precision	Recall	F1-Score	Support
Bintang	1.0000	1.0000	1.0000	3
Dafa	1.0000	1.0000	1.0000	3
Dyon	1.0000	1.0000	1.0000	3
Gadis	1.0000	1.0000	1.0000	1
Lintang	1.0000	1.0000	1.0000	3
Maliki	0.0000	0.0000	0.0000	1
Maria	1.0000	1.0000	1.0000	3
Radit	0.7500	1.0000	0.8571	3
Rival	1.0000	1.0000	1.0000	3
Ryan	1.0000	1.0000	1.0000	3
Sherly	1.0000	1.0000	1.0000	3
Syafiq	1.0000	1.0000	1.0000	3
Wahyu	1.0000	1.0000	1.0000	3
macro avg	0.9038	0.9231	0.9121	35
weighted avg	0.9500	0.9714	0.9592	35

Gambar 8. Classification Report

Gambar 8 menunjukkan *classification report* yang menyajikan *metrik precision, recall, F1-score*, dan *support* untuk setiap kelas. *Precision* menggambarkan tingkat ketepatan sistem dalam menetapkan identitas mahasiswa, *recall* menunjukkan kemampuan sistem dalam mengenali seluruh sampel wajah dari masing-masing mahasiswa, sedangkan *F1-score* merepresentasikan keseimbangan antara *precision* dan *recall*. Mayoritas kelas memiliki nilai *precision* dan *recall* sebesar 1,000, yang berarti sistem jarang menghasilkan kesalahan identifikasi maupun kehilangan pengenalan wajah. Beberapa kelas dengan jumlah sampel relatif sedikit, seperti Gadis dan Maliki, menunjukkan penurunan *F1-score* akibat kesalahan minor, namun *macro averaging* dan *weighted averaging* yang berada di atas 0,95 menunjukkan bahwa secara keseluruhan sistem presensi memiliki performa yang sangat baik untuk diterapkan.

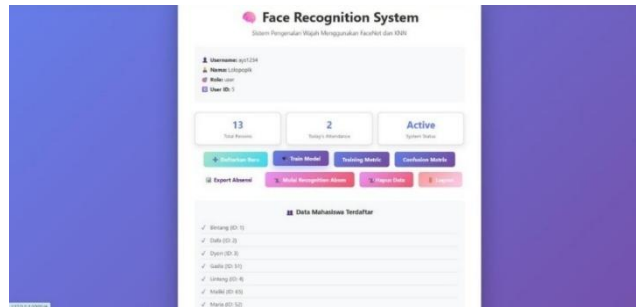
### 3.4 Implementasi Framework Python

Tahap implementasi sistem diimplementasikan dengan memanfaatkan bahasa pemrograman *Python* serta *framework Flask* sebagai layanan *backend* yang mengelola komunikasi antara antarmuka *web*, modul pemrosesan citra, dan basis data. Akuisisi citra wajah secara *real-time* dilakukan melalui kamera dan diproses menggunakan *OpenCV* untuk deteksi wajah serta prapemrosesan awal. Wajah yang terdeteksi kemudian diekstraksi fitur menggunakan model *FaceNet (CNN backbone)* yang menghasilkan *embedding* wajah, selanjutnya diklasifikasikan menggunakan algoritma *K-Nearest Neighbors (KNN)* berdasarkan perhitungan jarak. Hasil identifikasi diverifikasi menggunakan nilai *confidence*, dan apabila memenuhi ambang batas yang ditentukan, data kehadiran mahasiswa secara otomatis dicatat dan disimpan ke dalam basis data melalui antarmuka *web*, sehingga proses presensi dapat berlangsung secara *real-time*, terintegrasi, dan terdokumentasi dengan baik.



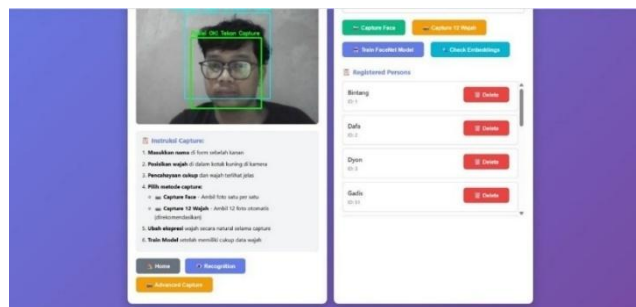
Gambar 9. Halaman Login

Gambar 9 menampilkan halaman *login*, pada halaman ini admin diminta memasukkan *username* dan *password* yang telah tersimpan di basis data. Halaman *login* ini memastikan hanya pihak pengguna yang memiliki hak akses yang bisa mengakses fungsi administrasi, sehingga keamanan data presensi dan data wajah mahasiswa tetap terjaga.



Gambar 10. Halaman Beranda

Gambar 10 menampilkan halaman beranda yang berfungsi sebagai pusat kontrol admin. Di dalamnya tersedia beberapa menu untuk mendaftarkan wajah baru, mengekspor data absensi, memulai proses pengenalan wajah, menghapus data, melihat daftar mahasiswa terdaftar, melakukan logout, serta menu untuk melatih ulang model dan melihat metrik evaluasinya.



Gambar 11. Halaman Daftar Baru

Gambar 11 pada menu daftar baru, sistem mengambil 12 foto wajah secara otomatis setelah tombol 'Capture' diklik. Foto-foto ini merekam variasi sudut dan ekspresi sebagai dataset pelatihan. Admin kemudian memasukkan nama mahasiswa, dan sistem menyimpan data wajah beserta identitasnya ke basis data untuk digunakan dalam proses pengenalan selanjutnya.



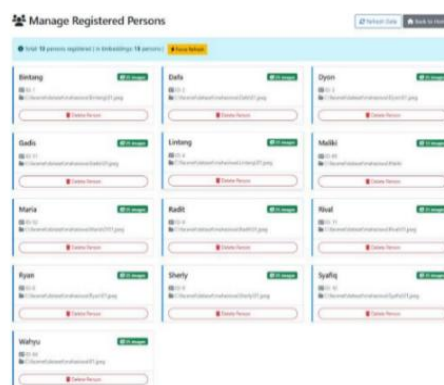
Gambar 12. Halaman Mulai Recognition Absen

Gambar 12 Halaman *Recognition* Absen, sistem melakukan deteksi dan identifikasi wajah berdasarkan data yang telah tersimpan di basis data. Setelah kecocokan terverifikasi, sistem secara otomatis mencatat kehadiran sebagai ‘masuk’ atau ‘pulang’ sesuai aturan waktu yang telah ditetapkan.



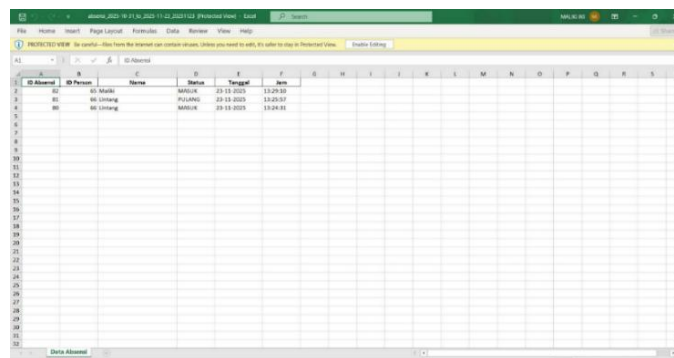
Gambar 13. Halaman *Eksport* Absensi

Gambar 13 ada halaman *eksport* absensi, di mana admin dapat mengunduh data kehadiran mahasiswa dalam format *Excel*, dengan data dapat difilter berdasarkan rentang tanggal tertentu untuk memudahkan analisis. Halaman ini juga menyediakan akses ke menu *registrasi person* serta menampilkan total mahasiswa yang telah terdaftar di sistem.



Gambar 14. Halaman *Regristasi Person*

Gambar 14 menampilkan halaman *registrasi person*, pada halaman ini menampilkan tampilan seluruh data mahasiswa yang sebelumnya tercatat dalam sistem yang dapat diakses oleh admin. Selain itu, admin memiliki kemampuan untuk menghapus data mahasiswa apabila diperlukan.



Gambar 15. Data Absensi Dalam *Excel*

---

Gambar 15 menunjukkan data absensi *Excel* yang di mana file yang diunduh berformat *Excel* dan berisi rekaman absensi mahasiswa, mencakup ID absensi, ID *person*, nama mahasiswa, status kehadiran ('masuk' atau 'pulang'), serta tanggal dan jam pada saat absensi.

## 4. Kesimpulan dan Saran

### 4.1 Kesimpulan

Berdasarkan hasil pengujian sistem presensi mahasiswa berbasis pengenalan wajah dengan model *FaceNet (CNN Backbone)* dan metode *KNN*, dapat disimpulkan bahwa:

1. Sistem presensi yang dikembangkan mampu mengidentifikasi wajah mahasiswa secara otomatis dan *real-time* dengan tingkat akurasi mencapai 97% pada dataset percobaan.
2. Model *FaceNet (CNN Backbone)* terbukti efektif dalam mengekstraksi fitur wajah yang representatif, sementara metode *KNN* mampu melakukan klasifikasi identitas secara sederhana namun akurat untuk kebutuhan presensi mahasiswa.
3. Performa sistem dipengaruhi oleh jumlah dan kualitas data latih, namun tetap mampu menangani variasi ekspresi dan pencahayaan.
4. Implementasi sistem ini berpotensi memberikan dampak positif bagi institusi pendidikan, khususnya dalam meningkatkan efisiensi dalam proses presensi, membatasi kecurangan mahasiswa dalam proses presensi, serta mendukung pengelolaan data kehadiran mahasiswa yang lebih akurat dan terintegrasi.

### 4.2 Saran

Berdasarkan hasil penelitian ini, beberapa saran dapat diberikan untuk pengembangan sistem ke depan yang dapat dijadikan referensi bagi penelitian dan pengembangan mendatang yaitu:

1. Menambah jumlah serta keragaman data latih, baik dari sisi jumlah mahasiswa maupun kondisi pengambilan citra, guna meningkatkan akurasi dan kemampuan generalisasi sistem.
2. Mengembangkan mekanisme keamanan tambahan seperti *anti-spoofing* (misalnya deteksi kedalaman wajah, kedipan mata, atau analisis tekstur) untuk mencegah penyalahgunaan sistem menggunakan foto atau video.
3. Mengintegrasikan sistem dengan teknologi pendukung lain, seperti sistem *multi-kamera*, basis data akademik, atau metode autentikasi tambahan, agar sistem lebih fleksibel dan siap diterapkan pada kelas dengan jumlah mahasiswa besar.
4. Melakukan evaluasi dan optimasi performa secara berkala untuk memastikan sistem tetap stabil, responsif, dan relevan dengan kebutuhan operasional institusi pendidikan.

## Daftar Pustaka

- [1] R. Gunawan, A. Maulana Yusuf, and L. Nopitasari, "Rancang Bangun Sistem Presensi Mahasiswa Dengan Menggunakan Qr Code Berbasis Android," vol. 14, no. 1, pp. 47–58, 2021, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom.page47>
- [2] Bambang Karyadi, "Pemanfaatan Kecerdasan Buatan Dalam Mendukung Pembelajaran Mandiri," 2023.
- [3] D. Indra, H. Herman, and F. S. Budi, "Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis Computer Vision," *Komputika : Jurnal Sistem Komputer*, vol. 12, no. 1, pp. 53–62, May 2023, doi: 10.34010/komputika.v12i1.9082.
- [4] T. Susim and C. Darujati, "Pengolahan Citra Untuk Pengenalan Wajah (*Face Recognition*) Menggunakan Opencv," *Jurnal Syntax Admiration*, vol. 2, no. 3, 2021.
- [5] Y. P. Manik and S. Sitohang, "Perancangan Sistem Absen Berbasis *Face Recognition*," *Jurnal Comasie*, vol. 12, no. 04, 2025.
- [6] S. R. Cholil, T. Handayani, R. Prathivi, and T. Ardianita, "Implementasi Algoritma Klasifikasi *K-Nearest Neighbor (KNN)* Untuk Klasifikasi Seleksi Penerima Beasiswa," 2021.
- [7] R. D. Artika, N. N. Rafifah, P. A. Dina, and C. Rozikin, "Perancangan *Data Pipeline* Untuk Analisis Pola Perjalanan dan Permintaan Layanan Transjakarta," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.8051.

- 
- [8] Angel Metanosa Afinda, "Machine Learning Workflow: Langkah-Langkah Praktis untuk Membangun Model AI," dicoding. Accessed: Nov. 20, 2025. [Online]. Available: <https://www.dicoding.com/blog/machine-learning-workflow-langkah-langkah-praktis-untuk-membangun-model-ai/>
- [9] Y. Azhar, A. Khoiriyah Firdausy, and P. J. Amelia, "Perbandingan Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Stroke," 2022, [Online]. Available: <https://doi.org/10.31598>
- [10] F. V. P. Samosir, L. P. Mustamu, E. D. Anggara, A. I. Wiyogo, and A. Widjaja, "Exploratory Data Analysis terhadap Kepadatan Penumpang Kereta Rel Listrik," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 2, Aug. 2021, doi: 10.28932/jutisi.v7i2.3700.
- [11] S. K. M. K. Prastyadi Wibawa Rahayu *et al.*, *Buku Ajar Data Mining*. 2023. Accessed: Nov. 22, 2025. [Online]. Available: [https://books.google.co.id/books?hl=id&lr=&id=LsOqEAAAQBAJ&oi=fnd&pg=PA59&dq=Data+Preprocessing+pengertian&ots=08BOuCJ5JC&sig=rUgbockSgU9HKp4RVbae7SPNO0&redir\\_esc=y#v=onepage&q=Data%20Preprocessing%20pengertian&f=false](https://books.google.co.id/books?hl=id&lr=&id=LsOqEAAAQBAJ&oi=fnd&pg=PA59&dq=Data+Preprocessing+pengertian&ots=08BOuCJ5JC&sig=rUgbockSgU9HKp4RVbae7SPNO0&redir_esc=y#v=onepage&q=Data%20Preprocessing%20pengertian&f=false)
- [12] Hasan Ismail Abdulmalik, "Deployment: Pengertian, Tujuan, dan Jenis-jenisnya." Accessed: Nov. 22, 2025. [Online]. Available: <https://www.dicoding.com/blog/deployment-pengertian-tujuan-dan-jenis-jenisnya/>
- [13] A. Valerian, D. Bernady, F. Wahyudi, J. P. Dinatha, and D. Barletyano, "Implementasi Unit Testing, Integration Testing, System Testing, dan Validation Testing Pada Aplikasi Berbasis Website (Studi Kasus: Kafe Saturdays)," 2025.
- [14] B. S. Gandhi, D. A. Megawaty, and D. Alita, "Aplikasi Monitoring dan Penentuan Peringkat Kelas Menggunakan Naïve Bayes Classifier," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 1, pp. 54–63, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>.