



## Rekayasa Perangkat Lunak Aplikasi Presensi *Mobile* Menggunakan Metode Deep Learning

Ragil Setiawan<sup>1</sup>, Nurcahya Pradana Taufik Prakisyah<sup>2\*</sup>, Rosihan Ari Yuana<sup>3</sup>  
<sup>1,2,3</sup> Pendidikan Teknik Informatika dan Komputer, FKIP, Universitas Sebelas Maret,  
Indonesia

Email: [mnurcahya.ptp@staff.uns.ac.id](mailto:mnurcahya.ptp@staff.uns.ac.id)

### ABSTRAK

Penelitian pengenalan wajah memiliki tantangan tersendiri dalam dunia penelitian, karena wajah merupakan hal yang kompleks, mulai dari ekspresi wajah, dan kondisi tertentu yang menjadikan pengenalan wajah sebagai eksperimen penelitian yang menarik. Banyak aplikasi yang berorientasi machine learning sudah beralih ke perangkat edge, tidak terkecuali dengan pengenalan wajah berbasis *mobile*. Melihat adanya geliat perkembangan algoritma pengenalan pola pada wajah seperti *naïve bayes* dan *backpropagation*, maka pada penelitian ini menggunakan model CNN seluler *MobileFaceNet* yang sedang populer untuk diimplementasikan pada aplikasi presensi pengenalan wajah berbasis *mobile* pada Pendidikan Teknik Informatika dan Komputer (PTIK) FKIP UNS. Metode deep learning ialah metode untuk memahami dan mengklasifikasi objek, dalam aplikasi yang dikembangkan yaitu sebuah wajah yang ditangkap dalam bentuk gambar.. Penelitian ini menggunakan bantuan *framework flutter* dan library *Tensorflow Lite* dalam pengembangan aplikasi presensi *mobile* pengenalan wajah secara *real-time*. Hasil dari penelitian ini untuk mengetahui nilai memorisasi dan generalisasi algoritma model CNN seluler *MobileFaceNet* pada aplikasi. Telah dilakukan uji coba sistem dengan melibatkan 30 orang relawan dalam pengujian memorisasi dan generalisasi yang berasal dari mahasiswa PTIK Angkatan 2016-2019 secara random sampling. Setiap pengujian dilakukan masing-masing sebanyak 10 iterasi. Hasil pengujian menunjukkan nilai memorisasi sistem sebesar 84.5% dan hasil generalisasi sebesar 70% dalam mengenali citra yang identik tapi tidak serupa dengan benar. Dari segi memorisasi maupun generalisasi, hasil ini lebih baik dibanding penelitian serupa dengan menggunakan *backpropagation*.

**Kata kunci:** *Deep learning, Framework flutter, Pengenalan wajah, Tensorflow Lite*

### ABSTRACT

*Facial recognition research has its challenges due to faces complexity, ranging from facial expressions and certain conditions that make facial recognition an exciting research experiment. Moreover, many-oriented applications of machine learning have moved to devices edge, and-based facial recognition is no exception mobile. Seeing the ongoing development of facial pattern recognition algorithms such as Viola Jones, Backpropagation, this research uses the MobileFaceNet mobile CNN model which is currently popular to be implemented in the mobile-based facial recognition presence application at the Information and Computer Engineering Education (PTIK) FKIP UNS. The deep learning method is a method for understanding and classifying objects. In the developed application, a face is captured in an image. This research uses the help of the flutter framework and the Tensorflow Lite library to develop a presence application mobile facial recognition in real-time. This paper aims to determine the value of the memorization and generalization algorithms model of CNN MobileFaceNet on the application. A trial of the system has been carried out by involving 30 volunteers in the testing from 2016-2019 PTIK students by random sampling. Each test was carried out for 10 iterations. From the test results, the system memorization value is 84.5%. On the other hand, the generalization results get 70% in*

*recognizing identical but not similar images correctly. In terms of memorization and generalization, these results are better than similar studies using backpropagation*

**Keywords:** *Deep learning, Face recognition, Framerowk flutter, Tensorflow Lite*

JIPTEK: Jurnal Ilmiah Pendidikan Teknik dan Kejuruan  
Vol 17 Issue 1 2024

DOI: <https://doi.org/10.20961/jiptek.v17i1.76556>

Received: 14/07/23 Revised: 09/09/23 Accepted: 21/09/23

Online: 14/12/23

© 2024 The Authors. Published by Universitas Sebelas Maret.  
This is an open access article under the CC BY license  
(<http://creativecommons.org/licenses/by/4.0/>).

## PENDAHULUAN

Penelitian pengenalan pola wajah manusia ialah salah satu bidang penelitian penting, dan penerapannya dapat diterapkan di berbagai metode dengan kelebihan dan kekurangan tertentu, hal tersebut dikarenakan wajah manusia merepresentasikan sesuatu yang kompleks (Harizahayu, 2021), pengenalan pola wajah dengan background, pencahayaan, ekspresi wajah, kondisi atau rintangan yang berbeda membuat perbedaan pendekatan untuk pengenalan pola wajah menjadi suatu eksperimen yang sangat menarik untuk dilakukan penelitian.

Pengenalan wajah (*Face recognition*) ialah masalah yang menantang dalam dunia penelitian dikarenakan beberapa faktor, salah satunya adalah banyaknya macam posisi gambar wajah pada pengenalan wajah itu sendiri. Pada pengambilan citra wajah, posisi wajah yang dapat ditangkap oleh kamera yaitu pada posisi depan, samping, maupun dari sudut tertentu yang menyebabkan terdapatnya bagian wajah seperti mata, hidung, maupun mulut tidak tertangkap secara keseluruhan. Faktor lain dari pengenalan wajah adalah ada atau tidaknya komponen struktural seperti kumis, jenggot, atau dengan atau tidak menggunakan kacamata pada

citra wajah, hal tersebut memiliki banyak variabilitas termasuk bentuk, warna, dan ukuran. Pencahayaan atau iluminasi, oklusi, dan ekspresi wajah juga merupakan faktor lain yang dapat mempengaruhi *Face recognition* atau pengenalan wajah (Wei & Li, 2013). Jaringan syaraf tiruan (JST) merupakan teknik yang dikenal dalam menangani masalah pengenalan wajah, dan juga dikenal memiliki kemampuan memorisasi yang merupakan kemampuan dalam memanggil kembali secara sempurna sebuah pola yang telah dipelajari sebelumnya, Generalisasi merupakan kemampuan dalam memberikan balasan yang dapat diterima terhadap pola-pola yang sebelumnya dipelajari serupa tapi tidak identik (Puspitaningrum, 2006).

Jika dibandingkan dengan algoritma lain seperti *naïve bayes* dan *K-nearest neighbour*, JST memiliki karakteristik uniknya sendiri yang berupaya untuk dapat meniru cara kerja otak manusia (Goodfellow, Bengio, & Courville, 2016). Deep learning ialah salah satu bentuk pendekatan dalam machine learning yang terinspirasi dari cara kinerja otak manusia dalam mempelajari maupun menyelesaikan suatu permasalahan (Skansi, 2018). Dalam JST terdapat beberapa jenis lapisan yang disebut *Multi Layer Perceptron* (MLP) yang sepenuhnya terhubung antara neuron dengan kemampuan klasifikasi yang kuat. MLP memiliki beberapa kekurangan atau masalah pada saat menginput citra, citra yang di masukan harus melakukan pre-processing, segmentasi, dan ekstraksi fitur

untuk mendapatkan performansi terbaik, karena MLP memiliki banyak parameter independen dalam arsitekturnya. Parameter independen dibentuk oleh skema yang sepenuhnya terhubung antara masukan dari lapisan yang sesuai dan peta fitur yang terhubung (Purnama, 2019).

*Convolution Neural Networks* (CNN) merupakan salah satu varian lain dari MLP yang terinspirasi oleh korteks visual pada mamalia. Model ini dapat mengurangi beberapa parameter independen, dan dapat menangani deformasi gambar masukan seperti translasi, rotasi dan penskala (Tran, Smith, & Dimla, 2019). Berdasarkan pada penjelasan singkat CNN tersebut, disimpulkan bahwa CNN dapat melakukan klasifikasi yang diperuntukan untuk data citra. Model CNN khusus untuk seluler atau smartphone, kelas CNN embedding fitur wajah yaitu *MobileFaceNets*. *MobileFaceNet* merupakan kelas model CNN yang menggunakan kurang dari 1 juta parameter dan dirancang khusus untuk verifikasi wajah secara *real-time* dengan akurasi tinggi pada perangkat seluler dibandingkan dengan model CNN seluler lainnya seperti *MobileNetV2* (Chen, Liu, Gao, & Han, 2018)

Banyak aplikasi yang berorientasi machine learning sudah beralih ke perangkat edge, tidak terkecuali dengan pengenalan wajah. *Edge* yaitu analisis lokal pada perangkat, artinya data ditangani secara lokal tanpa perlu mengunggah ke layanan cloud untuk melakukan pemrosesan lebih lanjut (Yangsila, 2023). Salah satu pemanfaatan pengenalan pola wajah pada perangkat *edge* adalah sistem kehadiran atau presensi. Presensi merupakan proses pencatatan

mahasiswa dalam menghadiri kelas (Radityatama, 2017). Adanya transformasi digital dalam sisi administratif, terutama di lingkungan Pendidikan Teknik Informatika dan Komputer FKIP UNS menjadikan perubahan proses presensi dari yang semula berbasis kertas dapat dikembangkan menjadi lebih modern dan terintegrasi melalui sebuah aplikasi presensi pengenalan wajah berbasis *mobile*.

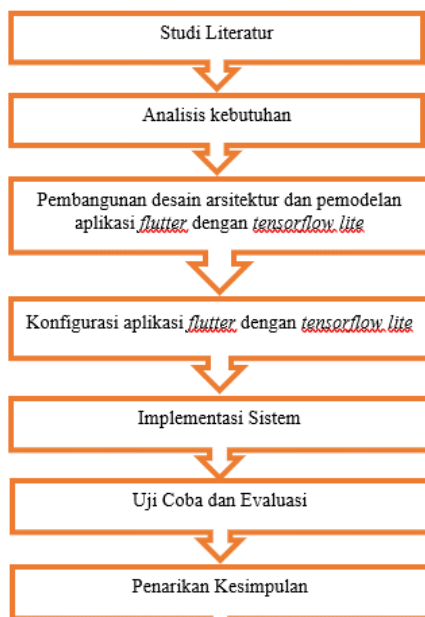
Melihat adanya kebutuhan bagi universitas dalam melakukan presensi bagi mahasiswanya dan didukung dengan lingkungan teknologi yang sudah memungkinkan implementasi *machine learning* dalam perangkat *edge*, maka penelitian ini melakukan pengembangan aplikasi presensi pengenalan wajah berbasis *mobile* menggunakan *framework Flutter* dengan bantuan library *Tensorflow Lite* untuk mengetahui nilai memorisasi dan generalisasi aplikasi, *Tensorflow Lite* diintegrasikan dengan *MobileFaceNet* untuk dapat melakukan verifikasi wajah di perangkat seluler.

Model CNN dipilih sebagai classifier pada aplikasi ini karena performanya yang lebih unggul dibanding algoritma lain seperti *gradient boosting tree*, *voting*, *Naïve Bayes*, dan *Hybrid random forest linear model* (Suneetha & Challa, 2023) maupun K-NN (Prakisya, Liantoni, Hatta, Aristyagama, & Setiawan, 2021). Model CNN juga tersedia pada *MobileFaceNet* sehingga aplikasi presensi melalui pengenalan wajah berbasis *mobile* dengan menggunakan *deep learning* diusulkan sebagai salah satu pemanfaatan kecerdasan buatan dalam dunia administratif pendidikan.

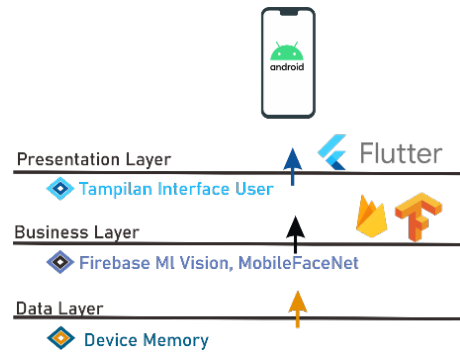
## METODE PENELITIAN

### Alur Penelitian

Tahap-tahap penelitian dilakukan mulai dari studi literatur yang dimana dilakukan pengumpulan referensi dan mengkaji terkait pengolahan pengenalan wajah berbasis *mobile* menggunakan flutter dan pengimplementasian *Tensorflow Lite* dengan permodelan metode deep learning. Tahapan kedua yang dilakukan yaitu analisis kebutuhan sistem sebagai gambaran umum rancangan aplikasi yang ingin dikembangkan, Tahapan ketiga setelah analisis kebutuhan dilakukan pembangunan desain arsitektur dan permodelan aplikasi flutter dengan *Tensorflow Lite* dengan 3 lapisan arsitektur yang terdiri dari *data layer*, *business layer*, dan *presentation layer* sehingga mendapatkan luaran aplikasi *mobile*. Arsitektur aplikasi dapat dilihat Gambar 2. Setelah melakukan arsitektur aplikasi dilakukan pemodelan aplikasi yang menggambarkan *interface* dari aplikasi.



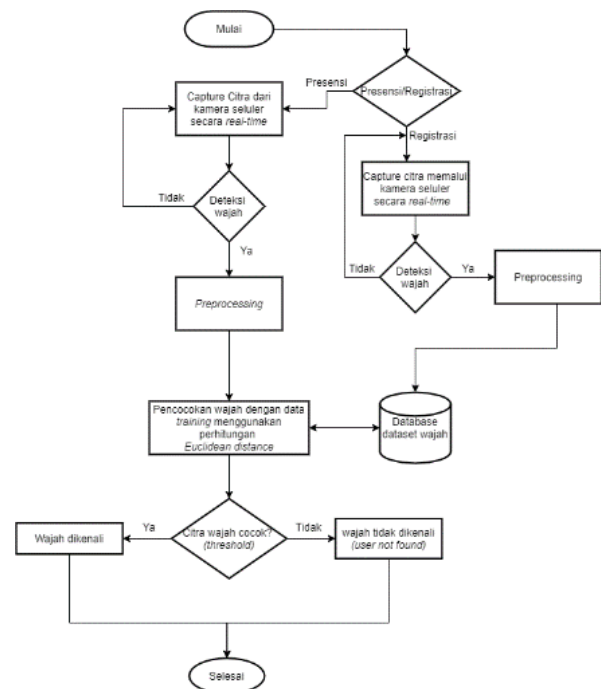
Gambar 1. Diagram Alir Penelitian



Gambar 2. Arsitektur aplikasi

### Desain Aplikasi

Tahapan-tahapan alur proses kerja sistem dimulai dari decision pilihan presensi atau melakukan registrasi wajah. Flowchart alur proses sistem dilihat pada Gambar 3 berikut :



Gambar 3. Flowchart Alur Proses Sistem

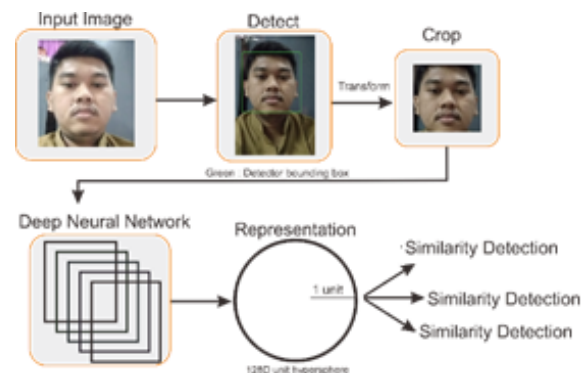
1. Capture citra dari kamera seluler secara *real-time*, tahap ini merupakan tahap awal untuk memperoleh citra untuk proses pengujian pengenalan wajah. Tahap ini menggunakan kamera seluler secara *real-time* dengan wajah menghadap depan dan berbagai ekspresi.

2. Deteksi wajah, setelah proses di atas selanjutnya mendeteksi wajah, tahap ini dilakukan mencari ada atau tidaknya wajah dalam suatu citra.
3. Tahap pre-processing, citra yang terdeteksi dapat diproses memberikan informasi yang lebih akurat, tahap ini akan melakukan transform vektor dan melakukan cropping. *Cropping* merupakan tahap pemotongan citra wajah yang bertujuan untuk mengambil bagian citra wajah yang diperlukan dan membuang bagian selain citra wajah.
4. Tahap selanjutnya merupakan pencocokan wajah dengan data training menggunakan perhitungan dengan *Euclidean distance*. Proses ini dibantu dengan melibatkan *MobileFaceNet* CNN untuk mencocokkan wajah yang sudah disimpan pada database berbentuk array JSON.
5. Jika wajah dikenali atau berada di atas angka *threshold*, maka sistem akan mengenali wajah dengan memberikan pesan ‘*welcome back username*’, jika wajah tidak dikenali sistem akan memberikan informasi ‘*user not found*’.

### Implementasi Deep Learning pada Aplikasi

Pada fase awal, sistem akan mendeteksi wajah dari masukan citra yang dilakukan, kemudian sistem akan melakukan cropping and alignment dari wajah yang terdeteksi, selanjutnya melakukan ekstraksi *feature vector* wajah yang berbentuk 128 array. Ekstraksi *feature vector* merupakan proses untuk mendapatkan informasi terkait wajah yang efektif dalam membedakan wajah-wajah orang yang berbeda dari citra wajah yang telah diselaraskan (Dewi & Ismawan,

2021). Hasil ekstraksi akan diproses untuk melakukan klustering, klasifikasi, dan deteksi kemiripan menggunakan perbandingan perhitungan *Euclidean distance* antara dua fitur vektor ekstraksi wajah. Gambar 4 menjelaskan alur dari penggunaan metode deep learning dalam melakukan verifikasi wajah.



Gambar 4. Alur Deep Learning Pada Aplikasi, Modifikasi Dari Rahouma dan MahFouz (2021)

### Penerapan Fitur Deteksi Keberadaan Wajah

Aplikasi yang dibangun menggunakan single shot multibox detector. Single shot multibox detector merupakan algoritma dalam face detection untuk memunculkan bounding box dimana pada dasarnya merupakan CNN berbasis *MobileFaceNet* dan telah disediakan langsung oleh *Tensorflow Lite* yang di dalamnya terdapat skor probabilitas kesesuaian wajah, ilustrasi implementasi pendeteksi area wajah pada aplikasi presensi dapat dilihat pada Gambar 5.

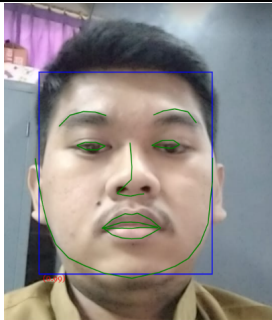



Gambar 5. Deteksi Keberadaan Wajah

## Kalibrasi Face Landmark Detection Dan Face Alignment

Setelah wajah terdeteksi, ketika melakukan capture citra sistem melakukan cropping wajah dan alignment sehingga wajah yang sudah dilakukan pemotongan akan diposisikan menjadi center. Sehingga sistem akan melakukan metode CNN sederhana yang menggunakan *MobileFaceNet* dalam menciptakan titik landmark wajah. Posisi titik-titik landmark wajah, wajah yang sudah dilakukan crop dapat diproses untuk memosisikannya ke center. Hasil dari kalibrasi face landmark detection dan alignment ini dapat dilihat pada Tabel 1.

Tabel 1. Kalibrasi Deteksi Landmark Wajah

<i>Face landmark detection simulation</i>	<i>Face alignment and cropping simulation</i>
	

## Penerapan Fitur Face Recognition

Setelah melakukan proses implementasi pertama dan kedua selanjutnya citra akan diproses oleh *MobileFaceNet* CNN untuk dilakukan ekstraksi data wajah. Dalam perbandingan dua wajah, deskriptor wajah yang terekstrak akan dibandingkan dengan dataset deskriptor wajah yang telah disimpan sebelumnya berupa array number pada database. Sistem akan menghitung menggunakan Eulidean distance antara dua deskriptor wajah dan menilai

apakah dua wajah tersebut serupa berdasarkan pada nilai threshold yang sudah ditentukan. Nilai threshold = 1.0 yang diimplementasikan pada sistem. Untuk menghitung eulidean distance dapat dilihat pada rumus dibawah ini :

$$d(q, p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (1)$$

Keterangan :

- $p_i$  = nilai vektor descriptor wajah ke pertama (dataset wajah)
- $q_i$  = nilai vektor deskriptor wajah kedua (masukan wajah)



## Pengujian dan Evaluasi Aplikasi

Pengujian dan evaluasi aplikasi dilakukan untuk menjalankan aplikasi yang sudah dikembangkan beserta pengujian kesesuaian terhadap kebutuhan aplikasi. Pada bagian uji coba aplikasi untuk mendapatkan nilai memorisasi dan generalisasi. Pengujian memorisasi dibutuhkan 20 orang relawan yang akan mengambil citra wajah, dan pengujian geralisasi dibutuhkan 10 relawan untuk mengambil citra wajahnya yang masing-masing dilakukan sebanyak 10 iterasi. Pengujian pengenalan dan verifikasi wajah dapat dilihat pada Tabel 2 dan Tabel 3 menggunakan skenario valid (benar) dan tidak valid (salah).

Pada simulasi pengujian, wajah dikategorikan ke dalam kelas benar, salah atau tidak dikenali dengan Eulidean distance. Apabila wajah yang dikonfigurasi oleh sistem di bawah angka threshold maka sistem akan mengenali wajah, jika di atas angka threshold

yang diberikan sistem akan memberikan informasi bahwa tidak ditemukan wajah serupa.

Tabel 2. Uji Verifikasi Wajah (valid)

Uji verifikasi wajah (scenario benar)	
Dataset wajah	
Data pembanding	
Hasil Pengenalan wajah	

Tabel 3. Uji Verifikasi Wajah (tidak valid)

Uji verifikasi wajah (scenario benar)	
Dataset wajah	
Data pembanding	
Hasil Pengenalan wajah	

Hasil uji coba pengenalan wajah akan dianalisis menggunakan perhitungan berikut :

$$M = \frac{p}{P} \times 100\% \mid G = \frac{p}{P} \times 100\% \quad (2)$$

Keterangan :

- M = Nilai Memorisasi
- G = Nilai Generalisasi
- p = jumlah benar wajah yang terdeteksi
- P = total iterasi wajah

Sebuah standar persentase kelayakan digunakan sebagai acuan dalam penentuan kriteria kelayakan sistem presensi dengan algoritma CNN seluler *MobileFaceNet* berdasarkan nilai memorisasi dan generalisasi Acuan skala persentase kelayakan dapat dilihat pada Tabel 4.

Tabel 4. Skala Persentase Kelayakan (Muhammad, R. Eka Murtinugraha, & Sittati Musalamah, 2020)

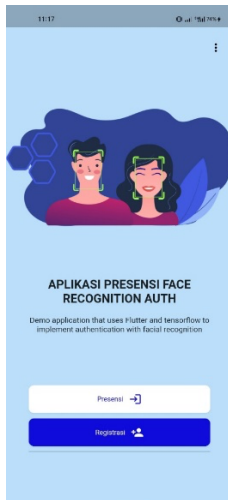
Persentase (%)	Interpretasi
81-100%	Sangat layak
61-80%	Layak
41-60%	Cukup Layak
21-40%	Tidak Layak
0-20%	Sangat Tidak Layak

## HASIL PENELITIAN DAN PEMBAHASAN

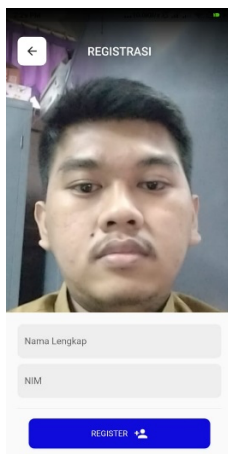
### Hasil Implementasi Antarmuka Pengguna

Penerapan *user interface* merupakan hasil tampilan dari tiap halaman yang muncul saat menjalankan aplikasi *mobile*. Aplikasi *mobile* ini dikembangkan menggunakan *Framework flutter* yang dikerjakan dan diimplementasikan pada lingkungan kerja *Visual Studio Code*. Implementasi *user interface* yang sudah dirender menjadi aplikasi android sudah sesuai dengan rancangan dan pengembangan yang telah

dilakukan. Gambar 6, Gambar 7 dan Gambar 8 merupakan *user interface* untuk *homescreen*, halaman registrasi dan halaman presensi berturut-turut.



Gambar 6. Halaman *HomeScreen*



Gambar 7. Halaman Registrasi



Gambar 8. Halaman Presensi

## Hasil Pengujian Prototype Aplikasi

Pengujian prototype aplikasi presensi *mobile* pengenalan wajah menggunakan metode *blackbox* testing oleh pakar untuk memastikan validitas aplikasi. Perolehan data dari evaluasi oleh expert review dapat dilihat pada Tabel 5.

## Hasil Uji Coba Memorisasi

Selain diuji pada dataset yang telah tersimpan, model juga diuji pada data tes yang belum pernah dikenali agar lebih valid dan terukur (Harjoko et al., 2018). Pada uji coba ini aplikasi akan melakukan pengujian terkait wajah yang sudah melakukan masukan dataset training pada sistem aplikasi.

Perhitungan nilai memorisasi dilakukan dengan membagi jumlah iterasi pengujian dengan jumlah yang dikenali dengan benar oleh aplikasi. Perhitungan jumlah dinyatakan benar ketika pengguna melakukan identifikasi wajah dan aplikasi dapat mengenali secara benar pengguna. Nilai dinyatakan salah ketika pengguna melakukan identifikasi wajah dan aplikasi tidak dapat mengenalnya atau memberikan keterangan “user not found” maupun pengguna yang melakukan *Face recognition* menghasilkan kesalahan output seperti terdeteksi bukan data dirinya sendiri melainkan data orang lain.

Dataset uji coba training merupakan hasil dari pengambilan citra responden yang melakukan registrasi wajah pada aplikasi untuk diuji. Kegiatan ini dibagi menjadi 4 skenario, yakni pencahayaan terang, pencahayaan kurang, jarak kamera kurang dari 30cm dan lebih dari 60cm. Tabel 6 – Tabel 9 merupakan hasil ujicoba masing-masing skenario.








Tabel 5. Deskripsi Hasil Pengujian

No.	Skenario Pengujian	Hasil Yang diharapkan	Hasil
<b>Halaman Menu Utama</b>			
1.	Melakukan klik tombol Presensi, Registrasi, dan <i>clear db</i>	Aplikasi menampilkan halaman Presensi, halaman Registrasi, dan menghapus <i>database</i>	Valid
<b>Halaman Presensi</b>			
1.	Melakukan pendeteksi wajah	Aplikasi menjalankan pendeteksi wajah dengan menampilkan <i>bounding box</i> di sekitar wajah	Valid
2.	Wajah tidak terdeteksi oleh <i>bounding box</i> atau tidak ada wajah pada halaman presensi	Aplikasi akan mengeluarkan pesan " <i>no face detection</i> "	Valid
3.	Menguji pengenalan wajah jika wajah sudah registrasi	Aplikasi mengenali wajah pengguna sesuai <i>dataset</i> dan menampilkan nama pengguna	Valid
4.	Menguji Pengenalan wajah jika wajah belum registrasi	Aplikasi akan menampilkan pesan " <i>user not found</i> "	Valid
5.	Masuk ke halaman selanjutnya dengan memasukkan NIM yang benar	Aplikasi menampilkan halaman berhasil melakukan presensi	Valid
6.	Maasuk ke halaman selanjutnya dengan NIM yang salah	Aplikasi menampilkan pesan "NIM kamu salah"	Valid
<b>Halaman Registrasi</b>			
1.	Melakukan deteksi wajah	Aplikasi menjalankan pendeteksi wajah dengan menampilkan <i>bounding box</i> (kotak hijau)	Valid
2.	Wajah tidak terdeteksi oleh <i>bounding box</i> atau tidak ada wajah pada halaman registrasi	Aplikasi akan mengeluarkan pesan " <i>no face detection</i> "	Valid
3.	Mengambil data wajah dengan mengklik tombol <i>capture</i>	Aplikasi mengambil data wajah yang akan disimpan di data <i>file json</i> pada <i>device memory</i>	Valid
4.	Menyimpan data wajah, nama, dan NIM	Aplikasi menyimpan <i>dataset</i> wajah, nama, dan NIM secara <i>real-time</i> . Jika berhasil maka ke halaman Utama	Valid






Tabel 6. Uji Coba Skenario 1

Nama	Gambar Saat Registrasi	Uji Pengambilan Gambar
Responden #1		Pencahayaan Terang
Responden #2		Pencahayaan Terang
Responden #3		Pencahayaan Terang
Responden #4		Pencahayaan Terang
Responden #5		Pencahayaan Terang

Tabel 7. Uji Coba Skenario 2

Nama	Gambar Saat Registrasi	Uji Pengambilan Gambar
Responden #6		Pencahayaan Kurang
Responden #7		Pencahayaan Kurang
Responden #8		Pencahayaan Kurang
Responden #9		Pencahayaan Kurang
Responden #10		Pencahayaan Kurang

Tabel 8. Uji Coba Skenario 3

Nama	Gambar Saat Registrasi	Uji Pengambilan Gambar
Responden #11		Jarak 30cm dari layar smartphone
Responden #12		Jarak 30cm dari layar smartphone
Responden #13		Jarak 30cm dari layar smartphone
Responden #14		Jarak 30cm dari layar smartphone
Responden #15		Jarak 30cm dari layar smartphone

Tabel 9. Uji Coba Skenario 4

Nama	Gambar Saat Registrasi	Uji Pengambilan Gambar
Responden #16		Jarak 60cm dari layar smartphone
Responden #17		Jarak 60cm dari layar smartphone
Responden #18		Jarak 60cm dari layar smartphone
Responden #19		Jarak 60cm dari layar smartphone
Responden #20		Jarak 60cm dari layar smartphone

Dari pengujian memorisasi yang dilakukan, skenario 1 mendapatkan nilai 94%, skenario 2 mendapatkan nilai 74%, skenario 3 mendapatkan nilai 84%, dan skenario 4 mendapatkan nilai 86%. Nilai memorisasi didapatkan dari kegiatan pengenalan wajah yang dilakukan 10 iterasi kepada masing-masing orang. Dari hasil tersebut dihitung kembali untuk mendapatkan nilai secara keseluruhan, perhitungan nilai rata-rata dilihat pada perhitungan dibawah ini :

$$\begin{aligned} \text{Memorisasi} &= (m_1+m_2+m_3+m_4)/4\% \\ &= (94+74+84+86)/4\% \\ &= 84.5\% \end{aligned}$$

### Hasil Uji Coba Generalisasi

Generalisasi adalah kemampuan untuk mendapatkan hasil respon yang dapat diterima oleh pola-pola masukan serupa tapi tidak identik terhadap yang sebelumnya dipelajari. Angka benar didapatkan dari iterasi yang dilakukan relawan sebanyak 10 iterasi, jumlah benar dinyatakan ketika aplikasi tidak mendeteksi dan memberikan keterangan user not found pada wajah relawan yang melakukan uji generalisasi, wajah salah didapatkan ketika aplikasi mendeteksi wajah user dan mengenalinya sebagai wajah yang sudah melakukan masukan dataset terlebih dahulu, sedangkan uji generalisasi dilakukan tanpa melakukan masukan dataset ataupun registrasi terlebih dahulu.

Pada uji coba generalisasi akan melakukan *Face recognition* secara *real-time* tanpa melakukan masukan dataset atau melakukan registrasi wajah pada aplikasi, terdapat 10 relawan yang akan melakukan uji

coba generalisasi dengan melakukan masing-masing 10 iterasi setiap orangnya. Mendapatkan hasil 70% dari hasil pengujiannya, perhitungan angka Generalisasi dijelaskan pada Tabel 10.

Table 10. Perhitungan Angka Generalisasi

No	Nama	Benar	Salah	Total
1	Responden #1	7	3	10
2	Responden #2	7	3	10
3	Responden #3	6	4	10
4	Responden #4	9	1	10
5	Responden #5	7	3	10
6	Responden #6	8	2	10
7	Responden #7	10	0	10
8	Responden #8	4	6	10
9	Responden #9	8	2	10
10	Responden #10	4	6	10
Jumlah		70	30	100
<b>Generalisasi</b>		<b>70%</b>	<b>30%</b>	<b>100%</b>

### Pembahasan Memorisasi dan Generalisasi

Dari seluruh hasil uji coba memorisasi dan generalisasi mendapatkan :

1. Ekspresi dan posisi wajah sangat mempengaruhi dalam melakukan *Face recognition* . Pada uji coba memorisasi aplikasi dari model 1 sampai dengan model 4 dari total 20 relawan mendapatkan sedikitnya 31 kesalahan dalam melakukan *Face recognition* dari total 200 iterasi yang dilakukan. Sedangkan dari generalisasi angka kesalahan sebanyak 30 dari 100 iterasi dan dilakukan 10 orang .
2. Intensitas cahaya yang terang pada dataset ataupun melakukan registrasi wajah, sangat berpengaruh terhadap memorisasi aplikasi. Intensitas cahaya yang baik mendapatkan

angka yang lebih tinggi dibandingkan intensitas cahaya yang kurang. Hal tersebut dilihat pada perbandingan model 1 dan model 2 dimana model 1 mendapatkan nilai memorisasi sebesar 94% sedangkan model 2 mendapatkan nilai 74%.

3. Spesifikasi kamera yang terdapat pada smartphone juga mempengaruhi angka memorisasi dan generalisasi. Hasil penangkapan citra dengan spesifikasi kamera yang bagus dapat menghasilkan citra yang lebih jelas sehingga data yang diperoleh lebih jelas dari citra kamera spesifikasi rendah dalam pengambilan citra.
4. Hasil uji memorisasi mendapatkan rata-rata sebesar 84,5% dan generalisasi sebesar 70%. Penilaian ini menunjukkan bahwa implementasi deep learning dapat dikatakan lebih baik dibanding penelitian serupa yang memanfaatkan algoritma *backpropagation* yang hanya menghasilkan nilai akurasi sebesar 67,36% saja (Abidin & Arifudin, 2012).

### SIMPULAN DAN SARAN

#### Simpulan

Perangkat lunak presensi wajah berbasis *mobile* dengan deep learning telah dapat dikembangkan. Teknologi yang digunakan pada aplikasi ini terdiri dari *framework flutter* untuk membangun UI dari aplikasi *mobile* dan *Tensorflow Lite* untuk mengimplementasikan machine learning dengan melibatkan *MobileFaceNet*. Konstruksi *backend* aplikasi presensi berbasis *mobile* menggunakan teknologi *flutter* dengan model CNN seluler dari *MobileFaceNet* Dalam ujicoba performa yang

dilakukan, aplikasi berhasil melakukan pendeteksi wajah sekaligus pengenalan wajah dengan rata-rata tingkat memorisasi sebesar 84.5%, dan tingkat generalisasi sebesar 70%. Berdasarkan rujukan pada Tabel 1, nilai memorisasi tergolong sangat layak dan generalisasi tergolong dalam kategori layak untuk digunakan dalam pembuatan aplikasi skala besar. Nilai akurasi yang besar merupakan landasan yang penting terutama bagi suatu aplikasi presensi mahasiswa ini agar terhindar dari misidentifikasi wajah yang berimbas pada kesalahan administratif pada laporan perkuliahan.

### Saran

Adapun terdapat beberapa hal untuk pengembangan aplikasi selanjutnya yaitu adanya integrasi dengan *firebase* sehingga aplikasi dapat menyimpan dataset ke *database server*. Dengan menambah integrasi data dengan *firebase*, maka catatan dataset wajah akan semakin berkembang dan proses pengenalan wajah dapat lebih valid. Selain itu, aplikasi dapat ditambahkan *liveness detector* sehingga masukan citra dari kamera dapat melakukan verifikasi apakah benar manusia atau foto wajah orang. Hal ini dapat dilakukan dengan memanfaatkan skor probabilitas buka tutup mata dan mulut pada deteksi wajah.

### DAFTAR PUSTAKA

Abidin, Z., & Arifudin, R. (2012). Pemanfaatan Biometrika Wajah Pada Sistem Presensi Menggunakan Backpropagation Neural Network. *Saintekno: Jurnal Sains Dan Teknologi*, 10(2). <https://journal.unnes.ac.id/nju/index.php/saintekno/article/view/5558>

- Chen, S., Liu, Y., Gao, X., & Han, Z. (2018). MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. <https://doi.org/10.48550/arXiv.1804.07573>
- 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>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Massachusetts: The MIT Press.
- Harizahayu, H. (2021). Pengenalan Ekspresi Raut Wajah Berbasis Jaringan Saraf Tiruan Backpropagation Dengan Metode Principal Component Analysis. *Barekeng: Jurnal Ilmu Matematika Dan Terapan*, 15(1), 037–046. <https://doi.org/10.30598/barekengvol15iss1pp037-046>
- Harjoko, A., Ratnaningsih, T., Suryani, E., Wiharto, Palgunadi, S., & Prakisyana, N. P. T. (2018). Classification of acute myeloid leukemia subtypes M1, M2 and M3 using active contour without edge segmentation and momentum backpropagation artificial neural network. *MATEC Web of Conferences*, 154. <https://doi.org/10.1051/mateconf/201815401041>
- Muhammad, H., R. Eka Murtinugraha, & Sittati Musalamah. (2020). Pengembangan Media Pembelajaran E-Learning Berbasis Moodle Pada Mata Kuliah Metodologi Penelitian. *Jurnal PenSil*, 9(1), 54–60. <https://doi.org/10.21009/jpensil.v9i1.13453>
- Prakisyana, N. P. T., Liantoni, F., Hatta, P., Aristyagama, Y. H., & Setiawan, A. (2021). Utilization of K-nearest neighbor algorithm for classification of white blood cells in AML M4, M5, and M7. *Open Engineering*, 11(1), 662–668. <https://doi.org/10.1515/eng-2021-0065>
- Purnama, B. (2019). *Pengantar Machine Learning - Konsep dan Praktikum dengan Contoh Latihan Berbasis R dan Python (1st ed.)*. Bandung: Penerbit Informatika.

- Puspitaningrum, D. (2006). Pengantar Jaringan Syaraf Tiruan (1st ed.). Yogyakarta: Andi.
- Radityatama, R. G. (2017). Rancang Bangun Aplikasi Mobile Android Sistem Kehadiran Mahasiswa Melalui Pencocokan Wajah dengan Menggunakan Library Android Face Recognition with Deep Learning Studi Kasus Jurusan Teknik Informatika ITS. Retrieved from <https://api.semanticscholar.org/CorpusID:64957924>
- Rahouma, K. H., & Mahfouz, A. Z. (2021). Design and Implementation of a Face Recognition System Based on API mobile vision and Normalized Features of Still Images. *Procedia Computer Science*, 194, 32–44. Elsevier B.V. <https://doi.org/10.1016/j.procs.2021.10.057>
- Skansi, S. (2018). *Introduction to Deep Learning*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-73004-2>
- Suneetha K. and Challa, K. and A. J. and R. Y. and K. S. (2023). Comparative Analysis on Heart Disease Prediction Using Convolutional Neural Network with Adapted Backpropagation. In R. and W. S.-J. and N. R. Rao B. Narendra Kumar and Balasubramanian (Ed.), *Intelligent Computing and Applications* (pp. 465–477). Singapore: Springer Nature Singapore. [https://doi.org/10.1007/978-981-19-4162-7\\_44](https://doi.org/10.1007/978-981-19-4162-7_44)
- Tran, H.-P., Smith, A., & Dimla, E. (2019). Offline Handwritten Text Recognition using Convolutional Recurrent Neural Network. 2019 International Conference on Advanced Computing and Applications (ACOMP), 51–56. <https://doi.org/10.1109/ACOMP.2019.00015>
- Wei, X., & Li, C.-T. (2013). Fixation and Saccade Based Face Recognition from Single Image per Person with Various Occlusions and Expressions. 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 70–75. <https://doi.org/10.1109/CVPRW.2013.18>
- Yangсила, C. (2023). Android Application for Face Recognition and Verification using MobileFaceNets. *Journal Of*

*Information Science And Technology*, 13(1), 1–9.  
<https://doi.org/10.14456/jist.2023.1>