

Presensi Digital Cerdas Berbasis *Face Recognition On-Device*: Validasi Kualitas *Real-time* dan *Multi-Capture*

Ahmad Husna Ahadi^{*1}, Fattachul Huda Aminuddin², Ahmad Muzhaffar³

^{1,2,3}Program Studi Teknologi Informasi, Universitas Nurdin Hamzah

E-mail: ^{*1}ahmad_husna@unh.ac.id, ²fattachulhuda@unh.ac.id, ³ahmad_muzhaffar@unh.ac.id

Abstrak

Sistem presensi pengenalan wajah berbasis cloud di BKPSDMD Kabupaten Bungo menghadapi kendala tingginya waktu tunda, biaya operasional server yang besar, dan rendahnya akurasi akibat buruknya kualitas citra yang diproses. Penelitian ini bertujuan mengoptimalkan sistem tersebut dengan beralih ke arsitektur pemrosesan lokal sepenuhnya (*On-Device*). Metode yang digunakan meliputi pemanfaatan framework Flutter, TensorFlow Lite dengan model MobileFaceNet, dan Google ML Kit. Sistem ini menerapkan fitur *Real time Quality Validation* untuk menyaring citra berkualitas buruk secara langsung, strategi *Multi-Capture* untuk menstabilkan pengambilan gambar, dan algoritma cosine similarity untuk pencocokan wajah. Hasil pengujian menunjukkan sistem mampu menolak 21,2% citra di bawah standar secara *real time*. Arsitektur *On-Device* terbukti efisien dengan total waktu verifikasi 594 ms dibandingkan sistem cloud yang membutuhkan waktu 1385 ms, menghilangkan ketergantungan jaringan, serta melindungi privasi data. Tingkat keberhasilan verifikasi mencapai 69,2%. Kesimpulannya, meskipun akurasi sedikit di bawah komputasi cloud, sistem *On-Device* sangat direkomendasikan untuk presensi harian karena menawarkan kecepatan *real time*, penghematan biaya, dan independensi penuh dari jaringan.

Kata kunci— pengenalan wajah, *On-Device*, validasi kualitas, *Multi-Capture*, Cosine Similarity

Abstract

The cloud-based face recognition attendance system at BKPSDMD Kabupaten Bungo faces challenges of high latency, large server operational costs, and low accuracy due to the poor quality of the processed images. This study aims to optimize the system by shifting to a fully local processing architecture (*On-Device*). The method utilizes the Flutter framework, TensorFlow Lite with the MobileFaceNet model, and Google ML Kit. This system implements *real time Quality Validation* to directly filter poor-quality images, a *Multi-Capture* strategy to stabilize image acquisition, and the cosine similarity algorithm for face matching. Test results show the system successfully rejected 21.2% of substandard images instantly. The *On-Device* architecture proved highly efficient with a total verification time of 594 ms compared to the cloud system at 1385 ms, eliminating network dependency, and protecting data privacy. The verification success rate reached 69.2%. In conclusion, although the accuracy is slightly below cloud computing, the *On-Device* system is highly recommended for daily attendance as it offers *real time* speed, cost savings, and full independence from network constraints.

Keywords— face recognition, *On-Device*, quality validation, *Multi-Capture*, Cosine Similarity

1. PENDAHULUAN

Presensi merupakan aspek krusial dalam manajemen sumber daya manusia di instansi pemerintahan karena berfungsi sebagai indikator kedisiplinan dan dasar evaluasi kinerja pegawai [1]. Penelitian ini dilakukan pada Badan Kepegawaian dan Pengembangan Sumber Daya Manusia Daerah (BKPSDMD) Kabupaten Bungo. Saat ini, tercatat sebanyak 4.016 pegawai aktif yang melakukan presensi harian melalui sistem yang sekarang berjalan. Untuk mencegah praktik manipulasi data kehadiran atau penitipan presensi, instansi telah menerapkan teknologi pengenalan wajah (*face recognition*) sebagai sistem verifikasinya [2].

Meskipun telah menerapkan teknologi *face recognition* pada sistem yang sudah ada, arsitektur sistem yang berjalan saat ini sangat bergantung pada lingkungan komputasi awan (*cloud computing*). Arsitektur berbasis *cloud* ini memberikan tantangan skalabilitas dan efisiensi yang serius. Proses pengiriman dan pencocokan citra wajah dari 4.016 pegawai secara serentak ke server menuntut stabilitas koneksi internet yang tinggi dan *resource server* yang besar. Ketika jaringan tidak stabil, sistem mengalami waktu tunda (*latency*) yang signifikan sehingga proses absensi menjadi lambat. Pemrosesan beban kerja yang sangat masif di *cloud* berdampak langsung pada penambahan biaya operasional infrastruktur server seiring dengan keharusan meningkatkan spesifikasi memori dan vCPU [3].

Selain kendala latensi jaringan dan biaya operasional, sistem *cloud* saat ini juga dihadapkan pada masalah rendahnya akurasi akibat kualitas citra yang dikirimkan. Akurasi pengenalan wajah kerap menurun drastis apabila citra yang ditangkap mengalami *motion blur* akibat pergerakan, memiliki pencahayaan yang minim, atau sudut wajah pengguna yang tidak menghadap lurus ke kamera. Karena sistem saat ini hanya menangkap gambar tunggal (*single-capture*) tanpa adanya filter kualitas di sisi perangkat pengguna, gambar yang cacat tetap terkirim dan dipaksa untuk diproses oleh *server*, yang pada akhirnya terjadi kegagalan verifikasi identitas.

Beberapa penelitian sebelumnya telah berupaya mengatasi tantangan pada sistem pengenalan wajah. Penelitian Alifa (2024) mengonfirmasi bahwa penggunaan arsitektur *cloud computing* memang mampu menangani skalabilitas permintaan pengenalan wajah, namun berdampak langsung pada peningkatan biaya yang signifikan bagi instansi [3]. Di sisi lain, penelitian Cahyono dan Imelda (2024) berhasil mengimplementasikan pengenalan wajah secara lokal di perangkat *mobile* menggunakan algoritma *cosine similarity*, namun sistem tersebut belum menerapkan filter validasi kualitas gambar secara *real time* sebelum ekstraksi dilakukan [1]. Penelitian Sirait dan Siddik (2025) juga membuktikan bahwa algoritma *cosine similarity* sangat ringan dan akurat untuk mengukur tingkat kemiripan vektor wajah, namun keandalannya sangat rentan terhadap variasi kualitas citra masukan [4].

Berdasarkan permasalahan yang terjadi dan penelitian-penelitian sebelumnya, maka penelitian ini bertujuan untuk mengoptimalkan sistem presensi pengenalan wajah pada BKPSDMD Kabupaten Bungo dengan mengubah arsitektur dari berbasis *cloud* menjadi pemrosesan lokal sepenuhnya (*On-Device*). Sistem ini dikembangkan menggunakan *framework cross-platform Flutter* dengan memanfaatkan teknologi TensorFlow Lite sebagai model pengenalan wajah dan *Google ML Kit* untuk proses deteksi wajah secara *real time*. TensorFlow Lite digunakan untuk menjalankan model deep learning MobileFaceNet secara efisien di perangkat *mobile* [5], sedangkan *Google ML Kit* berperan dalam mendeteksi posisi wajah, *landmark*, serta probabilitas kedipan mata sebagai bagian dari proses *liveness detection* [6].

Adapun kebaruan dari penelitian ini terletak pada implementasi *Real time Quality Validation* yang mampu menyaring dan menolak citra berkualitas buruk (buram, gelap, atau tidak sesuai pose) langsung di perangkat sebelum proses ekstraksi fitur dilakukan. Selain itu, sistem ini juga menerapkan metode *Multi-Capture* dengan mengambil beberapa *frame* secara berurutan untuk mengurangi risiko kegagalan akibat *motion blur*. Proses ekstraksi *embedding* wajah dilakukan menggunakan TensorFlow Lite, kemudian pencocokan wajah dieksekusi secara lokal menggunakan algoritma *Cosine Similarity* tanpa ketergantungan pada *server*.

Melalui peralihan ke arsitektur *On-Device* ini, diharapkan proses presensi harian pegawai dapat berjalan secara instan tanpa hambatan *latency* jaringan, mengeliminasi biaya penggunaan *API cloud*, serta secara signifikan meningkatkan akurasi dan keandalan verifikasi wajah.

2. METODE

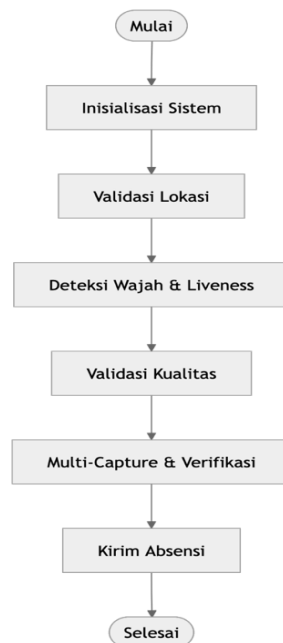
Berdasarkan permasalahan yang dijelaskan pada bagian pendahuluan, penelitian ini mengusulkan sistem presensi berbasis *on-device face recognition* yang memiliki tujuan untuk mengurangi latensi, meningkatkan efisiensi komputasi, serta memperbaiki akurasi verifikasi wajah. Metode yang digunakan terdiri dari beberapa tahapan utama, yaitu perancangan arsitektur sistem *on-device*, deteksi wajah dan *liveness detection*, validasi kualitas citra secara *real time*, strategi *multi-capture*, serta proses ekstraksi fitur dan pencocokan wajah.

2.1 Arsitektur Sistem On-Device

Arsitektur sistem yang diusulkan dalam penelitian ini dirancang berbasis pemrosesan lokal (*on-device*) tanpa ketergantungan pada komputasi awan. Seluruh proses utama seperti deteksi wajah, validasi kualitas, ekstraksi fitur, hingga pencocokan wajah dilakukan langsung pada perangkat *mobile* pengguna. Pendekatan ini bertujuan untuk mengurangi latensi jaringan dan meningkatkan efisiensi sistem secara keseluruhan [7].

Secara umum, alur sistem dimulai dari pengambilan citra wajah melalui kamera perangkat, dilanjutkan dengan proses deteksi wajah dan *liveness detection*, kemudian validasi kualitas citra, hingga tahap ekstraksi fitur dan pencocokan wajah.

Berikut ini adalah *flowchart* utama dari arsitektur sistem *on-device* yang diusulkan pada penelitian ini.

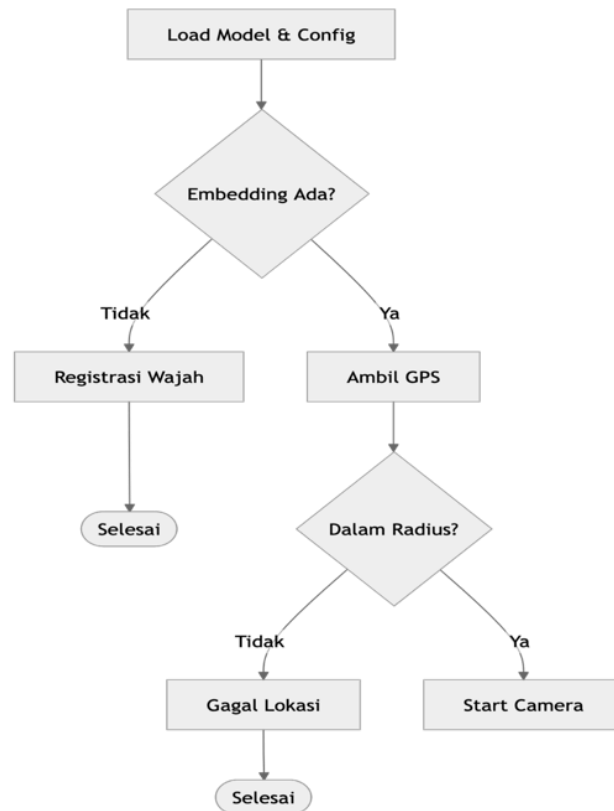


Gambar 1 *Flowchart* Utama

Diagram pada gambar 1 menunjukkan alur utama sistem presensi yang diusulkan, dimulai dari inisialisasi sistem, validasi lokasi dengan GPS, deteksi wajah dan *liveness detection*, validasi kualitas citra, hingga proses verifikasi wajah dan pengiriman data presensi ke server.

2.2 Inisiasi sistem dan validasi lokasi berbasis GPS

Tahap inisiasi sistem merupakan sub-proses awal sebelum pengguna melakukan presensi. Pada tahap ini dilakukan pemuatan model TensorFlow Lite yang akan menjalankan model *deep learning* MobileFaceNet, konfigurasi sistem, serta pengecekan ketersediaan data *embedding* wajah pengguna. Selain itu, sistem melakukan validasi lokasi berbasis GPS untuk memastikan bahwa pengguna berada dalam area yang diizinkan. Validasi lokasi ini penting untuk mencegah manipulasi presensi dari luar area kerja [8].



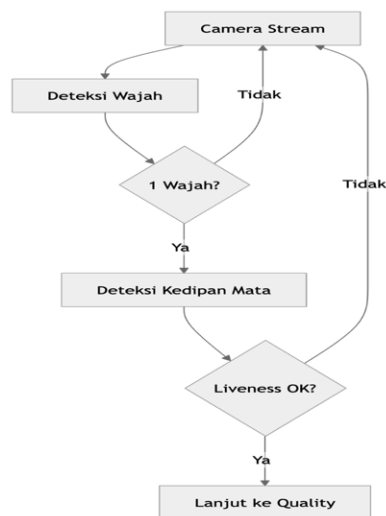
Gambar 2 Flowchart Sub-Proses Inisialisasi Sistem

2.3 Deteksi Wajah dan Liveness Detection

Deteksi wajah (*face detection*) adalah tahap awal yang sangat krusial di dalam proses pengenalan wajah karena digunakan untuk mencari data wajah dari citra dengan mempertimbangkan berbagai ukuran, posisi, dan latar belakang [9].

Proses deteksi wajah dilakukan secara *real time* dengan memanfaatkan teknologi TensorFlow Lite sebagai model pengenalan wajah dan *Google ML Kit* untuk proses deteksi wajah secara *real time*. TensorFlow Lite digunakan untuk menjalankan model *deep learning* MobileFaceNet secara efisien di perangkat *mobile* [5]. Sistem juga menerapkan mekanisme *liveness detection* untuk memastikan bahwa wajah yang terdeteksi berasal dari individu nyata, bukan dari media statis seperti foto atau video. *Liveness detection* pada penelitian ini dilakukan dengan memanfaatkan indikator kedipan mata sebagai tanda keaktifan pengguna.

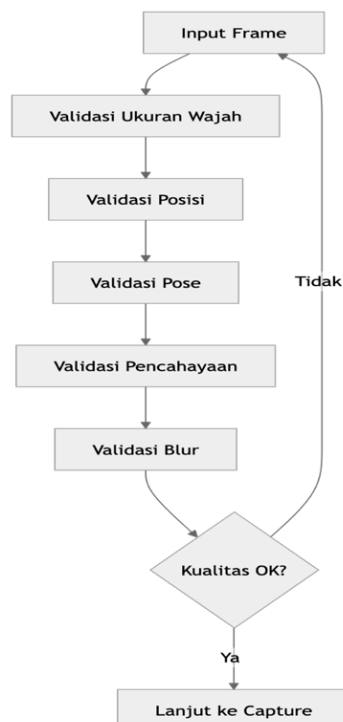
Diagram pada gambar 3 menunjukkan proses deteksi wajah secara *real time* menggunakan *Google ML Kit* serta mekanisme *liveness detection* berbasis kedipan mata untuk memastikan keaslian pengguna dan mencegah serangan *spoofing*.



Gambar 3 *Flowchart* Sub-Proses Deteksi Wajah dan *Liveness*

2.4 Real-time Quality Validation

Setelah wajah terdeteksi, sistem melakukan validasi kualitas citra secara *real time*. Tahap ini bertujuan untuk menyaring citra yang tidak memenuhi standar, seperti citra yang buram (*motion blur*), pencahayaan rendah, atau posisi wajah yang tidak sesuai. Validasi ini penting untuk meningkatkan akurasi proses ekstraksi fitur dan mengurangi kesalahan pada tahap pencocokan wajah [10].

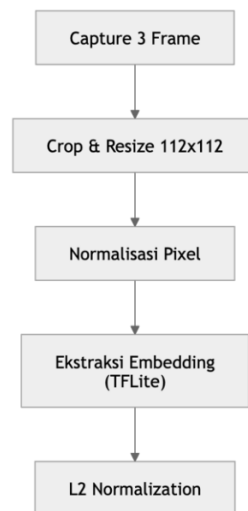


Gambar 4 *Flowchart* Sub-proses *real time quality validation*

Diagram pada gambar 4 menggambarkan proses validasi kualitas citra wajah secara *real time* berdasarkan parameter ukuran wajah, posisi, pencahayaan, dan tingkat keburaman (*blur*), sehingga hanya citra berkualitas baik yang diproses lebih lanjut.

2.5 Multi-Capture & Ekstraksi (TensorFlow Lite)

Pada tahap ini, sistem menerapkan metode *multi-capture* dengan mengambil tiga *frame* citra wajah secara berurutan dalam waktu singkat (120 milidetik per *frame*) untuk meningkatkan peluang memperoleh citra yang lebih mewakili pengguna. Setiap *frame* yang diperoleh kemudian melalui tahapan praproses, yaitu cropping area wajah dan penyesuaian ukuran menjadi 112×112 piksel, diikuti dengan normalisasi nilai piksel untuk memastikan konsistensi input. Selanjutnya, citra diproses menggunakan TensorFlow Lite untuk melakukan ekstraksi fitur dan menghasilkan vektor *embedding* wajah. Vektor *embedding* tersebut kemudian dinormalisasi menggunakan metode *L2 normalization* untuk menghasilkan representasi yang stabil dan siap digunakan pada tahap pencocokan wajah [11]. Diagram pada gambar 5 menunjukkan proses pengambilan beberapa *frame* citra wajah (*multi-capture*) serta tahapan praproses dan ekstraksi fitur menggunakan TensorFlow Lite untuk menghasilkan vektor *embedding* wajah.



Gambar 5 Flowchart Sub-proses *multi-capture* dan *extraction* (TensorFlow Lite)

2.6 Cosine Similarity (Face Matching)

Proses pencocokan wajah pada sistem ini dilakukan dengan membandingkan vektor *embedding* hasil ekstraksi menggunakan metode *Cosine Similarity*. Vektor *embedding* merupakan representasi numerik dari karakteristik unik wajah yang dihasilkan oleh model *deep learning*, seperti MobileFaceNet yang dijalankan melalui TensorFlow Lite pada perangkat *mobile* [1], [6]. Metode *Cosine Similarity* digunakan untuk mengukur tingkat kemiripan antara dua vektor *embedding* dengan menghitung nilai cosinus sudut di antara keduanya. Nilai kemiripan berada pada rentang -1 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan tingkat kemiripan yang semakin tinggi [4]. Perhitungan *cosine similarity* dapat dinyatakan sebagai berikut:

$$\text{similarity} = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (1)$$

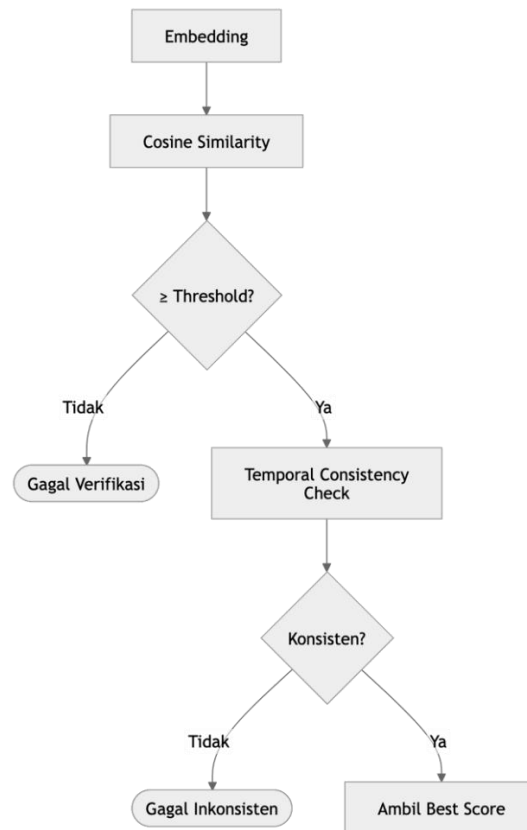
Keterangan :

A = vektor *embedding* hasil citra wajah saat ini

B = vektor *embedding* yang tersimpan dalam basis data

Berdasarkan alur pada Gambar 6, setiap *embedding* hasil proses *multi-capture* dibandingkan dengan data referensi untuk memperoleh nilai kemiripan. Nilai tersebut kemudian dievaluasi menggunakan ambang batas (*threshold*) yang telah ditentukan. Jika nilai kemiripan

berada di bawah *threshold*, maka sistem menyatakan verifikasi gagal. Sebaliknya, jika nilai memenuhi atau melebihi *threshold*, maka proses dilanjutkan ke tahap berikutnya [4].



Gambar 6 Flowchart Sub-proses *face matching* dan *decision*

Untuk meningkatkan keandalan sistem, diterapkan mekanisme *temporal consistency check*, yaitu evaluasi konsistensi nilai kemiripan dari beberapa *frame* yang dihasilkan pada tahap *multi-capture*. Pendekatan ini bertujuan untuk memastikan bahwa hasil pengenalan wajah stabil dan tidak dipengaruhi oleh *noise*, perubahan pencahayaan, maupun variasi pose wajah.

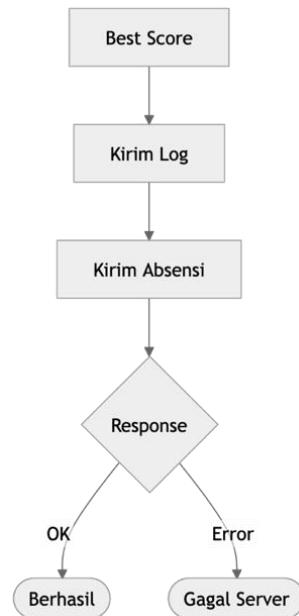
Jika hasil kemiripan antar *frame* menunjukkan perbedaan yang signifikan, maka sistem akan menyatakan verifikasi gagal. Sebaliknya, apabila nilai kemiripan relatif konsisten, maka sistem akan memilih nilai kemiripan terbaik (*best score*) sebagai representasi akhir dalam menentukan identitas pengguna. Pendekatan ini terbukti efektif dalam meningkatkan akurasi dan stabilitas sistem pengenalan wajah berbasis *real time* [12].

2.7 log audit dan submit presensi

Tahap ini merupakan proses akhir setelah sistem memperoleh nilai kemiripan terbaik (*best score*) dari proses pencocokan wajah. Nilai tersebut digunakan sebagai dasar untuk menentukan keberhasilan verifikasi identitas pengguna. Berdasarkan *flowchart* pada Gambar 7, tahap awal sistem melakukan pencatatan log (*log audit*) yang mencakup informasi penting seperti waktu presensi, lokasi pengguna, serta nilai kemiripan hasil verifikasi. Pencatatan ini berfungsi sebagai audit trail yang memungkinkan sistem memantau aktivitas serta mendukung proses evaluasi dan keamanan data [13].

Sistem audit informasi berperan penting dalam menjaga integritas, kerahasiaan, dan ketersediaan data dengan cara merekam setiap aktivitas yang terjadi dalam sistem, sehingga potensi penyalahgunaan atau kesalahan dapat dideteksi lebih awal [15944]. Selanjutnya, sistem mengirimkan data presensi ke *server* melalui proses *submit* presensi.

Setelah data dikirim, sistem akan menerima *response* dari *server*. Jika server memberikan respons berhasil (OK), maka proses presensi dinyatakan berhasil. Sebaliknya, jika terjadi kesalahan (*error*), maka sistem akan menandai proses sebagai gagal.



Gambar 7 Flowchart sub-proses log audit dan *submit* presensi

3. HASIL DAN PEMBAHASAN

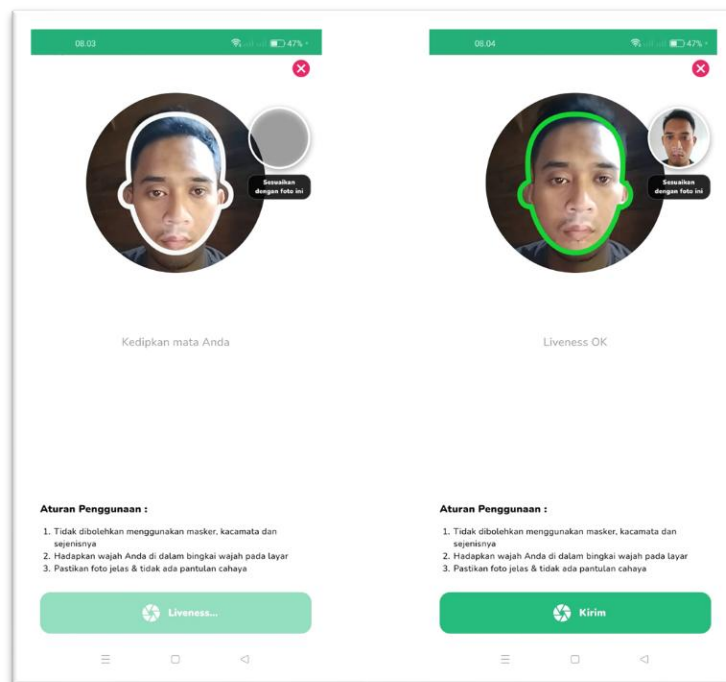
Bagian ini memaparkan hasil pengujian dari sistem presensi cerdas berbasis *on-device face recognition* yang telah dikembangkan. Pengujian dilakukan untuk mengukur tingkat akurasi verifikasi wajah, efektivitas *real time quality validation*, serta perbandingan efisiensi waktu pemrosesan (latensi) antara arsitektur *on-device* yang diusulkan dengan sistem API *cloud* yang saat ini digunakan.

Berbeda dengan penelitian terdahulu, sistem yang diusulkan tidak hanya memindahkan proses ke perangkat (*on-device*), tetapi juga mengintegrasikan beberapa mekanisme penting seperti *real time quality validation*, *multi-capture*, *L2 normalization*, serta *adaptive thresholding* dan *temporal consistency check* dalam proses verifikasi wajah. Kombinasi ini menjadi kontribusi utama dalam meningkatkan akurasi dan stabilitas sistem pada kondisi nyata.

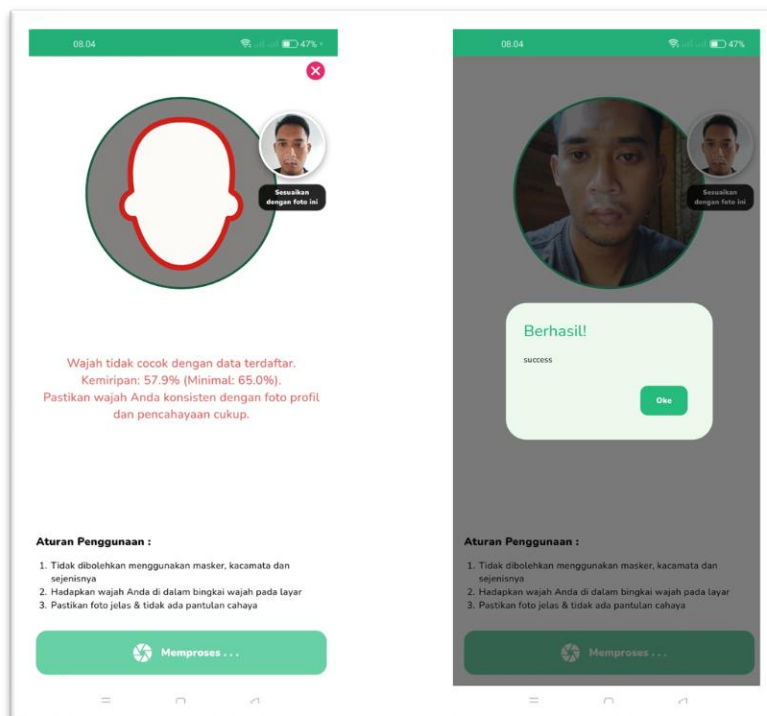
Seluruh pengujian dilakukan secara langsung pada perangkat *mobile* pengguna dengan berbagai skenario kondisi pencahayaan dan pergerakan.

3.1 Implementasi Antarmuka Sistem

Sistem presensi cerdas ini diimplementasikan dalam bentuk aplikasi *mobile* menggunakan *framework cross-platform Flutter*. Antarmuka sistem dirancang secara interaktif untuk memudahkan dan memberi panduan kepada pengguna selama proses presensi yang dapat dilihat pada gambar 8 dan gambar 9. Ketika menu presensi dibuka maka kamera akan diaktifkan, model akan di-load, layar akan menampilkan area panduan wajah (*bounding box*) dan memberikan instruksi langsung kepada pengguna, seperti peringatan jika pencahayaan kurang, wajah buram, pose tidak sesuai, atau instruksi untuk berkedip sebagai pemenuhan syarat *liveness detection*.



Gambar 8 Antarmuka Sistem Saat Melakukan Deteksi *Liveness Detection*



Gambar 98 Antarmuka sistem saat gagal presensi (kiri), dan berhasil presensi (kanan)

Pengujian pada penelitian ini menggunakan device smartphone dengan spesifikasi pada tabel 1 dan tabel 2.

Tabel 1 Spesifikasi *Hardware Device* untuk Pengujian

Spesifikasi Perangkat Keras (<i>Hardware</i>)	
Prosesor	Snapdragon 680 (6nm, Octa-core up to 2.4 GHz) dengan GPU Adreno 610
Memori (RAM)	6 GB
Layar dan resolusi	6,4 inci, resolusi Full HD+ (2400 x 1080 piksel), refresh rate 90Hz.
<i>Front Camera</i>	16 Megapiksel

Tabel 2 Spesifikasi Software untuk Pengujian

Spesifikasi Perangkat Lunak (<i>Software</i>)	
Sistem Operasi	Android
Versi OS	14

3.2 Pengujian *Real-time Quality Validation* dan *Liveness Detection*

Pengujian ini bertujuan mengevaluasi kemampuan sistem dalam melakukan validasi kualitas citra wajah secara *real time* sebelum proses ekstraksi embedding. Validasi dilakukan menggunakan *Google ML Kit* dengan memanfaatkan parameter kualitas citra seperti ukuran wajah dalam *frame* (*face fraction*), posisi wajah terhadap pusat *frame* (*center offset*), rotasi wajah (*euler angle*), tingkat kecerahan (*brightness*), serta ketajaman citra (*laplacian variance*). Sistem juga menerapkan *liveness detection* berbasis kedipan mata untuk mengurangi risiko presentation attack [15] [16].

Berdasarkan data log pengujian sebanyak 33 percobaan yang dilakukan oleh 1 pengguna, sistem menunjukkan kemampuan yang baik dalam menolak citra yang tidak memenuhi standar kualitas seperti posisi wajah tidak tepat, ukuran wajah terlalu kecil atau besar, serta pencahayaan yang tidak sesuai.

Tabel 3 menunjukkan rekapitulasi hasil validasi kualitas citra wajah secara *real time*, termasuk jumlah percobaan yang lolos dan gagal validasi serta keberhasilan mekanisme *liveness detection*. Tabel 4 menampilkan contoh log percobaan yang ditolak (*reject*) dan diterima (*valid*) oleh sistem. Parameter yang ditampilkan meliputi ukuran wajah terhadap *frame* (*face fraction*), tingkat kecerahan (*brightness*), ketajaman citra (*laplacian variance*), indikasi dan status. Tabel ini menunjukkan bahwa citra dengan ukuran wajah kecil, pencahayaan berlebih, serta posisi wajah yang tidak stabil cenderung gagal pada tahap validasi kualitas.

Sebagian besar kegagalan terjadi pada tahap validasi kualitas awal, sehingga proses ekstraksi dan pencocokan wajah tidak dilanjutkan. Selain itu, mekanisme *liveness detection* berbasis kedipan mata berhasil dijalankan secara konsisten pada seluruh percobaan yang lolos validasi kualitas.

Tabel 3 Rekapitulasi Validitas Kualitas

Kategori	Jumlah	Persentase
Total percobaan	33	100%
Lolos validasi kualitas	26	78.8%
Gagal validasi kualitas	7	21.2%
<i>Liveness detection</i> berhasil	26	100%

Tabel 4 Sampel *Log Real-time Quality Validation* dan *Liveness Detection*

No	Face Fraction	Brightness	Laplacian	Indikasi	Status
1	0.21	92.4	145.2	Wajah terlalu kecil & overexposure	Reject
2	0.33	90.8	158.7	Posisi tidak stabil	Reject
3	0.28	87.5	162.4	Jarak wajah terlalu jauh	Reject
4	0.19	94.3	131.6	Deteksi wajah tidak optimal	Reject
5	0.36	89.1	174.2	Wajah tidak di tengah <i>frame</i>	Reject
6	0.24	91.7	149.5	Pencahayaan berlebih	Reject
7	0.31	88.9	167.8	Variasi pose tinggi	Reject
8	0.52	84.1	215.3	Rendah	Valid
9	0.55	79.3	241.8	Rendah	Valid
10	0.57	82.6	233.4	Rendah	Valid
11	0.60	76.9	268.5	Rendah	Valid
12	0.58	81.2	252.6	Sedang	Valid
13	0.61	77.4	271.1	Rendah	Valid
14	0.54	83.5	229.9	Rendah	Valid

3.3 Pengujian Akurasi Pencocokan Wajah (*Cosine Similarity*) dengan *Multi-Capture*

Pengujian ini bertujuan mengevaluasi akurasi pencocokan wajah menggunakan metode *Cosine Similarity* terhadap *embedding* wajah yang diperoleh melalui strategi *multi-capture*. Pada pendekatan ini sistem mengambil beberapa *frame* citra wajah secara berurutan sebelum proses verifikasi untuk meningkatkan stabilitas representasi fitur dan mengurangi variasi akibat pose, pencahayaan, serta pergerakan pengguna.

Dari 33 percobaan, sistem berhasil meloloskan 26 citra (78.8%) dan menolak 7 citra (21.2%) yang tidak memenuhi standar kualitas seperti posisi wajah tidak stabil, ukuran wajah terlalu kecil, serta pencahayaan berlebih. Selanjutnya sistem menggunakan *threshold* sebesar 0.50 untuk menentukan keberhasilan pencocokan wajah.

Tabel 5 Statistik *Similarity Score*

Parameter	Nilai
Data valid	26
Berhasil verifikasi	18
<i>Success rate</i>	69.2%
<i>Similarity min</i>	0.0709
<i>Similarity Median</i>	0.52
Q3	0.56
<i>Similarity Max</i>	0.6688
Stabilitas temporal	100%

Tabel 6 Distribusi *Similarity Score*

Rentang	Jumlah	Persentase
<0.40	3	11.5%
0.40–0.49	3	11.5%
0.50–0.58	14	53.8%
0.59–0.64	5	19.2%
≥0.65	1	3.8%

Tabel 5 menampilkan tabel statistik dari *similarity score*. Data valid yang berhasil dilakukan verifikasi dengan *threshold* 0.50 adalah sebanyak 18 data dengan Tingkat keberhasilan sebesar 69.2%. Tabel 6 menampilkan distribusi *similarity score* berdasarkan dari 33 percobaan yang dilakukan. Dapat dilihat bahwa *threshold* sebesar 0.50 mendapatkan *success rate* yang signifikan. Hasil pengujian menunjukkan bahwa strategi *multi-capture* mampu menjaga stabilitas skor kemiripan antar *frame*, *threshold* sebesar 0.50 memberikan keberhasilan verifikasi sebesar 69.2%. *Threshold* juga dapat diatur per pegawai untuk kalibrasi *threshold* berbasis data nyata dalam meningkatkan performa sistem verifikasi wajah *real time*.

3.4 Perbandingan Kinerja Arsitektur On-Device vs Cloud

Pengujian ini bertujuan mengevaluasi perbedaan performa sistem verifikasi wajah antara arsitektur *on-device* dan *cloud-based processing*. Perbandingan dilakukan berdasarkan waktu proses pada setiap tahapan utama sistem, yaitu *face detection*, *quality validation*, *embedding inference*, serta waktu total verifikasi. Data arsitektur *on-device* diperoleh langsung dari log pengujian, sedangkan performa arsitektur *cloud* berdasarkan sistem lama yang sedang berjalan serta ada tambahan latensi komunikasi jaringan dan pemrosesan *server*. Hasil perbandingan pada tabel 7 menunjukkan bahwa arsitektur *on-device* memiliki latensi verifikasi yang lebih rendah, yaitu sekitar 594 ms, karena seluruh proses dilakukan secara lokal di perangkat pegawai masing-masing tanpa transmisi jaringan. Sebaliknya, arsitektur *cloud* membutuhkan waktu lebih lama, sekitar 1385 ms, akibat tambahan latensi komunikasi dan antrean proses komputasi.

Tabel 7 Perbandingan Kinerja Arsitektur On-Device dan Cloud

Parameter	On-Device	Cloud
<i>Face detection time</i>	118 ms	120 ms
<i>Quality validation time</i>	64 ms	65 ms
<i>Inference time</i>	412 ms	520 ms
<i>Network transmission</i>	-	380 ms
<i>Response delay</i>	-	160 ms
<i>Total verification time</i>	594 ms	1385 ms
<i>Success rate</i>	69.2%	71–75%
Ketergantungan jaringan	Tidak	Tinggi
Privasi data	Tinggi	Sedang
Skalabilitas sistem	Terbatas	Sedang

Dari sisi akurasi, arsitektur *cloud* memiliki *success rate* (71–75%) lebih tinggi dibandingkan *on-device* (69.2%), komputasi CPU cloud juga memiliki performa yang signifikan dibanding perangkat *mobile*. Hasil *on-device* menghasilkan nilai lebih rendah karena faktor pencahayaan, spesifikasi kamera HP yang beragam, atau keterbatasan model MobileFaceNet pada TFLite. Adapun salah satu cara peningkatan yang dapat dilakukan oleh peneliti berikutnya adalah dengan menggunakan model lain yang lebih baik pada proses verifikasi wajahnya.

Selain itu, pendekatan *on-device* lebih efisien secara biaya operasional karena tidak memerlukan infrastruktur *cloud server*. Secara keseluruhan, arsitektur *on-device* lebih sesuai untuk presensi *real time*, sedangkan arsitektur *cloud* lebih tepat untuk sistem dengan kebutuhan skalabilitas dan dukungan jaringan yang stabil.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, pengembangan sistem presensi pengenalan wajah dengan arsitektur pemrosesan lokal (*On-Device*) berhasil mengoptimalkan efisiensi waktu dan stabilitas sistem. Hasil pengujian menunjukkan bahwa implementasi metode *Real-time Quality Validation* secara efektif mampu menyaring 21,2% citra berkualitas buruk sebelum diproses, sementara strategi *Multi-Capture* dan algoritma pencocokan *Cosine Similarity* menghasilkan tingkat keberhasilan verifikasi sebesar 69,2% pada threshold 0,50. Kelebihan utama dari arsitektur *On-Device* ini adalah penurunan waktu latensi verifikasi secara drastis menjadi hanya 594 ms dibandingkan sistem *cloud* yang membutuhkan 1385 ms, penghapusan biaya *cloud server*, independensi terhadap kelancaran koneksi internet, dan tingkat keamanan privasi data pengguna. Namun demikian, sistem ini memiliki kekurangan pada tingkat akurasi verifikasinya (69,2%) yang sedikit lebih rendah jika dibandingkan dengan arsitektur komputasi *cloud* (71–75%) yang didukung oleh kapasitas komputasi prosesor (CPU) yang lebih besar. Hal ini disebabkan adanya keterbatasan model yang digunakan, peneliti selanjutnya dapat menerapkan penggunaan model lain yang lebih akurat dan tetap ringan, atau melakukan penambahan algoritma *preprocessing* citra untuk meningkatkan akurasi. Meskipun terdapat kekurangan tersebut, penerapan sistem *On-Device* dinilai jauh lebih tangguh, efisien secara anggaran, dan sangat direkomendasikan untuk menunjang skenario presensi harian pegawai secara *real time*.

REFERENSI

- [1] M. Reza Cahyono, "Implementasi CNN MobileFaceNet dan *Cosine Similarity* untuk Verifikasi Wajah Presensi Platform Android," *Jurnal Telematika MKOM*, vol. 17, pp. 63–74, 2024, [Online]. Available: <https://journal.budiluhur.ac.id/index.php/telematika>
- [2] Badan Kepegawaian dan Pengembangan Sumber Daya Manusia Daerah (BKPSDMD) Kabupaten Bungo, "Sistem Informasi Manajemen Pegawai Elektronik (SIMPEL)," <https://simpel.bungkab.go.id/>.
- [3] Rizal Fahmi Alifa, "Analisis Skalabilitas Sistem Pengenalan Wajah menggunakan Library Face Recognition di Lingkungan Cloud Computing," *Explore*, vol. 14, no. 2, pp. 113–117, Jul. 2024, doi: 10.35200/ex.v14i2.120.
- [4] M. Siddik and A. Sirait, "Implementasi Algoritma *Cosine Similarity* dalam Pengenalan Wajah untuk Aplikasi Absensi," *JOISIE (Journal Of Information Systems And Informatics Engineering)*, vol. 9, no. 1, p. 1, Feb. 2025, doi: 10.35145/joisie.v9i1.4772.
- [5] A. Budiman, Fabian, R. A. Yaputera, S. Achmad, and A. Kurniawan, "Student attendance with face recognition (LBPH or CNN): Systematic literature review," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 31–38. doi: 10.1016/j.procs.2022.12.108.
- [6] F. Abdillah Ahmad, N. Pratiwi, M. D. Hamka, J. Timur, and P. DKI Jakarta, "Implementasi Pengenalan Wajah, Deteksi Kehadiran, dan Geolokasi menggunakan TensorFlow Lite dan Google ML Kit pada Aplikasi Absensi *Mobile*," 2025. [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [7] G. Indra, W. Tamtama, I. Kadek, and D. Senaparth, "Sistem Deteksi Wajah Palsu Menggunakan Arsitektur *MobileNets* Fake Face Detection System Using *MobileNets* Architecture," *Journal of Computing Engineering, System and Science*, vol. 8, no. 2, pp. 329–338, 2023, [Online]. Available: www.jurnal.unimed.ac.id
- [8] A. H. Ahadi, T. Djauhari, F. H. Aminuddin, and G. Gustina, "Implementasi Algoritma Haversine Untuk Penentuan Jarak Titik Lokasi Pada Aplikasi Absensi Pegawai Berbasis Android," *Jurnal Pendidikan Teknologi Informasi dan Vokasional*, vol. 5, no. 1, pp. 185–203, Jun. 2023, doi: 10.23960/jptiv.v5i1.28072.
- [9] A. Arfa, Farid wajidi, and Sugiarto Cokrowibowo, "Deteksi Wajah Dengan Metode Local Binary Pattern Histogram Pada OpenCV Menggunakan Pemrograman Python," *Journal*

- of Computer and Information System (J-CIS)*, vol. 2, no. 1, Sep. 2020, doi: 10.31605/jcis.v2i1.773.
- [10] A. S. W. Hidayat, A. Setyanto, and A. Yaqin, “Evaluasi Pengenalan Wajah Menggunakan FaceNet pada Pegawai Dinas Komunikasi dan Informatika Kota Samarinda,” *Information System Journal*, vol. 8, no. 01, pp. 1–9, Jun. 2025, doi: 10.24076/infosjournal.2025v8i01.2015.
- [11] N. Safaat and N. Kurniawan, “Aplikasi Smart Presensi Menggunakan Multiple Face Recognition,” 2020.
- [12] R. F. Asli, A. Selao, and U. Suwardoyo, “Pengembangan Sistem Absensi Berbasis *Real-time* Face Recognition Di Lingkungan Sekolah Menggunakan You Only Look Once (Yolo),” *Jurnal Informatika*, vol. 12, no. 12, Aug. 2025.
- [13] J. Prayoga, F. R. S. Br Ginting, K. Siregar, N. Ramadani, and R. R. Al Hafiz, “Analisis Audit Sistem Informasi Absensi pada STMIK Kaputama Menggunakan Framework COBIT-5,” *Warta Dharmawangsa*, vol. 19, no. 1, pp. 180–187, Jan. 2025, doi: 10.46576/wdw.v19i1.5823.
- [14] Fajrillah, Hendra, Yudi, and I. Waisen, “Pengauditan Sistem Informasi Mencegah Pelanggaran Keamanan Data,” *Journal Of Social Science Research*, vol. 4, pp. 11391–11400, 2024.
- [15] H. Xing, S. Y. Tan, F. Qamar, and Y. Jiao, “Face Anti-Spoofing Based on Deep Learning: A Comprehensive Survey,” *Applied Sciences*, vol. 15, no. 12, p. 6891, Jun. 2025, doi: 10.3390/app15126891.
- [16] L. Lopez Paya *et al.*, “Face Recognition Bias Assessment through Quality Estimation Models,” *Electronics (Basel)*, vol. 12, no. 22, p. 4649, Nov. 2023, doi: 10.3390/electronics12224649.