ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

Implementasi Sistem Pengenalan Wajah dengan Antarmuka untuk Pendataan Kehadiran Berbasis Python

Daffa Risky Pratama[1], Mohammad Idhom[2*], Eka Prakarsa Mandyartha[3]

[1,3] Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[1]19081010052@upnjatim.ac.id
[3]eka_prakarsa.fik@upnjatim.ac.id
[2] Sains Data, Universitas Pembangunan Nasional "Veteran" Jawa Timur
**Corresponding author email*: idhom@upnjatim.ac.id

Abstrak*— Perkembangan teknologi informasi dan kecerdasan buatan telah memberikan dampak signifikan terhadap berbagai aspek kehidupan manusia, termasuk dalam bidang administrasi dan manajemen kehadiran. Penelitian ini bertujuan untuk mengimplementasikan sistem dan antarmuka aplikasi pengenalan wajah berbasis Python guna mendukung proses pendataan kehadiran peserta video meeting secara otomatis. Sistem dibangun menggunakan algoritma Convolutional Neural Network (CNN) dengan model FaceNet untuk tahap pengenalan wajah, serta algoritma Haarcascade sebagai metode deteksi wajah awal. Aplikasi dilengkapi dengan Graphical User Interface (GUI) berbasis pustaka customtkinter agar seluruh fungsi sistem dapat dioperasikan melalui tampilan interaktif tanpa perintah berbasis teks. Metodologi yang digunakan adalah Software Development Life Cycle (SDLC) dengan model Incremental, yang membagi implementasi ke dalam empat modul utama, yaitu: (1) modul deteksi wajah, (2) modul pengenalan wajah, (3) modul antarmuka pengguna (GUI), dan (4) modul penyimpanan data. Hasil implementasi menunjukkan bahwa sistem berfungsi secara operasional untuk mendeteksi, mengenali, dan mencatat identitas wajah dari citra hasil tangkapan layar video meeting. Pengujian pada satu frame Zoom yang berisi 15 wajah menghasilkan precision sebesar 90,9%, recall 71,4%, dan accuracy 66,7%. Hasil tersebut menunjukkan bahwa sistem mampu mengenali identitas peserta dengan baik pada kondisi visual yang memadai serta memberikan kemudahan penggunaan melalui antarmuka GUI yang responsif dan terintegrasi.

Kata Kunci— **Pengenalan wajah, Python, CNN, FaceNet, GUI, OpenCV, Deep Learning**

I. PENDAHULUAN

Kemajuan teknologi informasi dan komunikasi saat ini mendorong penerapan sistem digital dalam berbagai bidang kehidupan. Salah satu aspek yang turut mengalami transformasi adalah sistem pendataan kehadiran. Proses absensi konvensional yang dilakukan secara manual masih sering menghadapi permasalahan seperti human error, keterlambatan pencatatan, serta manipulasi data kehadiran[1].

Pada masa pascapandemi, aktivitas daring melalui *video meeting* semakin banyak digunakan baik di lingkungan akademik maupun profesional[2]. Namun, sistem pendataan kehadiran untuk pertemuan daring masih sering dilakukan secara manual, seperti melalui daftar hadir atau konfirmasi lisan, yang menimbulkan potensi kesalahan administrasi dan kurang efisien dalam waktu. Oleh karena itu, dibutuhkan sistem yang mampu melakukan pendataan kehadiran secara otomatis tanpa intervensi langsung dari pengguna[3][4].

Teknologi pengenalan wajah (*face recognition*) merupakan salah satu solusi yang efektif karena mampu mengidentifikasi identitas seseorang berdasarkan ciri visual yang unik pada wajahnya. Pengenalan wajah memiliki keunggulan dibandingkan metode lain seperti sidik jari atau kode QR karena tidak memerlukan kontak fisik dan dapat diterapkan secara real-time.

Sistem pengenalan wajah secara umum terdiri dari dua tahap utama, yaitu deteksi wajah (*face detection*) dan pengenalan wajah (*face recognition*). Deteksi wajah berfungsi untuk menemukan lokasi wajah pada citra, sedangkan pengenalan wajah bertugas untuk mencocokkan identitas berdasarkan data wajah yang telah tersimpan sebelumnya[5][6]. Penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN) dengan model *FaceNet* untuk tahap pengenalan wajah, serta algoritma Haarcascade untuk deteksi wajah awal.

Selain itu, aplikasi ini dibuat agar mudah digunakan oleh pengguna dengan memanfaatkan pustaka *customtkinter* untuk membangun *Graphical User Interface* (GUI) yang interaktif dan sederhana. Pendekatan *Incremental Model* dalam metode SDLC diterapkan agar setiap modul sistem dapat dikembangkan dan diuji secara bertahap, sehingga menghasilkan kerangka sistem yang stabil dan fleksibel untuk dikembangkan lebih lanjut.

II. TINJAUAN PUSTAKA

A. *Deteksi Wajah*

Deteksi wajah merupakan tahap awal dalam sistem pengenalan wajah yang berfungsi untuk menemukan posisi wajah manusia dalam suatu citra atau video. Salah satu metode deteksi yang paling umum digunakan adalah algoritma Haarcascade yang dikembangkan oleh Paul Viola dan Michael Jones[7]. Algoritma ini bekerja berdasarkan penggunaan *Haar-like features* dan *Integral Image* yang mempercepat proses pendeteksian dengan memanfaatkan pola kontras antara area terang dan gelap pada wajah manusia.

Haarcascade memanfaatkan konsep *Cascade Classifier*, yaitu rangkaian filter yang diterapkan secara bertahap untuk menyaring area citra yang berpotensi mengandung wajah[8]. Dengan proses ini, sistem dapat mendeteksi wajah frontal dengan cepat dan akurat tanpa membutuhkan daya komputasi

ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

tinggi. Dalam penelitian ini, Haarcascade digunakan sebagai metode utama untuk mendeteksi wajah dari tangkapan layar (*screenshot*) video meeting.

B. Pengenalan Wajah

Pengenalan wajah adalah proses identifikasi individu berdasarkan pola dan fitur unik pada wajahnya. Proses ini biasanya dilakukan setelah sistem mendeteksi wajah dalam citra. Salah satu metode yang digunakan dalam penelitian ini adalah *FaceNet*, model yang dikembangkan oleh Google yang memanfaatkan jaringan saraf tiruan *Deep Convolutional Neural Network (CNN)*.

FaceNet mengubah wajah menjadi *embedding vector*, yaitu representasi numerik berdimensi tinggi yang menggambarkan ciri-ciri unik wajah. Dengan menggunakan fungsi *triplet loss*, FaceNet mampu meminimalkan jarak antara embedding wajah yang sama dan memaksimalkan jarak antara wajah yang berbeda. Dalam sistem ini, setiap wajah yang terekam akan dibandingkan dengan data wajah yang tersimpan di *database JSON* menggunakan perhitungan jarak *Euclidean*. Jika jarak embedding berada di bawah ambang batas tertentu (*threshold*), maka wajah dianggap cocok dengan identitas yang tersimpan.

C. Convolutional Nerual Network (CNN)

CNN merupakan arsitektur jaringan saraf tiruan yang dirancang khusus untuk pengolahan citra. CNN terdiri dari beberapa lapisan utama, yaitu *convolution layer*, *pooling layer*, dan *fully connected layer* [9]. Lapisan konvolusi berfungsi mengekstraksi fitur dari citra, sedangkan lapisan pooling mengurangi dimensi fitur untuk mempercepat proses dan mengurangi redundansi data.

CNN bekerja dengan cara mempelajari pola visual yang berulang seperti tepi, tekstur, dan bentuk wajah. Dalam konteks pengenalan wajah, CNN dapat digunakan untuk mempelajari fitur kompleks yang sulit dikenali oleh metode konvensional seperti LBPH atau Eigenfaces. CNN juga dapat melakukan generalisasi terhadap variasi ekspresi, sudut pandang, dan pencahayaan wajah.

D. Incremental Model

Model pengembangan perangkat lunak *Incremental* merupakan salah satu pendekatan dari *Software Development Life Cycle (SDLC)* yang membagi proses pengembangan menjadi beberapa tahap (increment). Setiap tahap menghasilkan komponen sistem yang berfungsi sebagian dan kemudian dikembangkan lebih lanjut hingga menjadi sistem yang lengkap.

Pendekatan ini digunakan dalam penelitian untuk mengimplementasikan empat modul utama sistem, yaitu: (1) modul deteksi wajah, (2) modul pengenalan wajah, (3) modul antarmuka pengguna (GUI), dan (4) modul penyimpanan data. Setiap modul dikembangkan dan diuji secara bertahap untuk memastikan fungsionalitas dan stabilitas sistem sebelum diintegrasikan secara keseluruhan.

E. Graphical User Interface (GUI)

GUI adalah antarmuka visual yang mempermudah pengguna dalam berinteraksi dengan sistem. Dengan menggunakan pustaka *customtkinter*, aplikasi ini memiliki tampilan modern dan intuitif. GUI memungkinkan pengguna mengoperasikan fungsi aplikasi seperti mengambil tangkapan layar, memperbarui database wajah, dan menyimpan hasil pengenalan tanpa menulis kode.

F. Python dan OpenCV

Python merupakan bahasa pemrograman tingkat tinggi yang mendukung berbagai pustaka untuk *machine learning* dan *computer vision* [10]. Salah satu pustaka yang digunakan dalam penelitian ini adalah OpenCV, yang menyediakan berbagai fungsi pengolahan citra seperti *face detection*, *feature extraction*, dan *image transformation*.

Kombinasi Python dan OpenCV memudahkan proses pengembangan sistem pengenalan wajah karena memiliki dukungan komunitas luas, dokumentasi lengkap, dan kompatibilitas tinggi dengan pustaka lain seperti TensorFlow dan Keras.

G. Tkinter dan customtkinter

Tkinter merupakan pustaka standar Python untuk pengembangan *Graphical User Interface (GUI)*. Pada penelitian ini digunakan *customtkinter*, versi modern dari Tkinter yang mendukung tampilan bertema *dark mode*, elemen melengkung (*rounded buttons*), serta desain yang lebih interaktif dan estetis.

GUI memiliki peran penting dalam sistem yang diimplementasikan karena menjadi penghubung antara pengguna dengan seluruh proses utama. Melalui GUI, pengguna dapat menambah data wajah, memulai proses deteksi dan pengenalan, memperbarui database, serta menampilkan hasil pengenalan secara langsung. Desain antarmuka dibuat agar mudah digunakan oleh pengguna non-teknis dengan navigasi yang sederhana dan tampilan yang konsisten.

III. METODOLOGI

Penelitian ini menggunakan pendekatan *Software Development Life Cycle (SDLC)* dengan model *Incremental*. Model ini memungkinkan pengembangan aplikasi dilakukan secara bertahap, di mana setiap modul diuji dan disempurnakan sebelum diintegrasikan ke dalam sistem keseluruhan.

Aplikasi dirancang untuk mendeteksi dan mengenali wajah peserta *video meeting* dari gambar hasil tangkapan layar (*screenshot*) yang diambil oleh pengguna. Proses sistem dibagi ke dalam empat modul utama, yaitu: modul deteksi wajah, modul pengenalan wajah, modul antarmuka pengguna (GUI), dan modul penyimpanan data.

Flowchart desain sistem ini dapat dilihat pada Gambar 1. Berdasarkan fungsi utamanya, sistem yang dirancang dalam penelitian ini hanya membutuhkan satu perangkat untuk dijalankan. Perangkat ini akan bekerja sebagai pengambil tangkapan layar di *video meeting* juga sekaligus mengelola data tersebut melalui aplikasi dan memberikan hasil pengenalan wajah. Gambar 1 menjelaskan proses kerja sistem yang dimulai dari pengambilan tangkapan layar. Pengambilan tangkapan layar ini dilakukan dengan mengklik tombol yang telah disediakan di aplikasi. Aplikasi akan mengambil tangkapan layar dari *video meeting* yang berisi daftar peserta. Kemudian hasil tangkapan itu disimpan aplikasi untuk dilakukan proses pengenalan wajah dengan membandingkan terhadap data-data yang sudah ada.

ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

Gbr. 1 *Flowchart sistem pengenalan wajah*

A. Analisa Kebutuhan

Tahap ini mencakup analisis terhadap kebutuhan perangkat keras, perangkat lunak, serta kebutuhan fungsional sistem. Sistem dijalankan pada komputer dengan spesifikasi minimal prosesor multi-core, RAM 8 GB, serta mendukung pustaka Python. Dalam penelitian digunakan perangkat dengan CPU *AMD Ryzen 5 7500F, GPU NVIDIA RTX 4060, RAM 32 GB DDR5,* dan sistem operasi *Windows 10 64-bit.* Kebutuhan perangkat lunak meliputi bahasa pemrograman *Python 3.x* dengan pustaka: *OpenCV, TensorFlow, Keras, PIL*, *customtkinter*, dan *JSON*. *Editor* yang digunakan adalah *Visual Studio Code*.

Secara fungsional, sistem diharapkan dapat: mengambil tangkapan layar (*screenshot*) dari jendela *video meeting*, mendeteksi area wajah dari hasil tangkapan layar menggunakan algoritma Haarcascade, mengenali identitas wajah yang terdeteksi dengan membandingkan embedding wajah terhadap data yang tersimpan di file JSON, menampilkan hasil pengenalan di antarmuka GUI serta menyimpan hasil ke file *txt* atau *CSV* secara otomatis.

B. Perancangan Sistem

Tahap ini menjelaskan rancangan dari keempat modul utama sistem, yaitu deteksi wajah, pengenalan wajah, antarmuka pengguna, dan penyimpanan data hasil kehadiran. Flowchart *Increment Model* yang digunakan pada penelitian ini dapat dilihat pada Gambar 2.

1) Modul Deteksi Wajah: Modul deteksi wajah berfungsi untuk mendeteksi area wajah pada gambar hasil tangkapan layar *video meeting*. Algoritma yang digunakan adalah Haarcascade dari pustaka OpenCV. Proses dimulai dengan membaca gambar, mengonversinya menjadi skala abu-abu (*grayscale*), lalu menerapkan fungsi *detectMultiScale()* untuk mendeteksi wajah frontal.

Parameter *scaleFactor* diatur sebesar 1.1 dan *minNeighbors* sebesar 4 untuk menjaga keseimbangan antara kecepatan dan akurasi deteksi. Area wajah yang terdeteksi diberi tanda persegi, kemudian hasilnya dikirim ke modul pengenalan wajah.

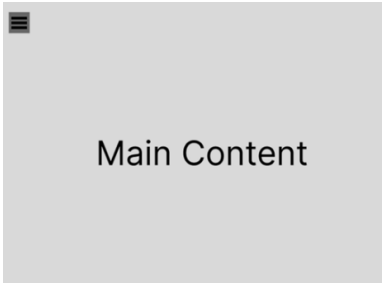
Gbr. 2 *Flowchart Increment Model yang digunakan*

2) Modul Pengenalan wajah*:* Tahap pengenalan wajah dilakukan menggunakan model *FaceNet* berbasis *Convolutional Neural Network (CNN)*. Model ini menghasilkan *face embedding*, yaitu representasi numerik berdimensi tinggi dari wajah.

Hasil embedding dibandingkan dengan data wajah yang telah tersimpan pada file face_sign.json. Pencocokan dilakukan menggunakan jarak *Euclidean* antara embedding baru dan data referensi. Jika nilai jarak di bawah ambang batas (*threshold*), maka sistem mengenali wajah tersebut dan menampilkan identitas peserta yang sesuai pada GUI.

3) Modul Antarmuka Pengguna (GUI): Modul Antarmuka Pengguna (*Graphical User Interface*, GUI) pada aplikasi ini berperan sebagai penghubung antara pengguna dengan sistem. GUI dikembangkan menggunakan pustaka customtkinter, yang mendukung pembuatan antarmuka modern dan responsif dibandingkan dengan Tkinter standar.

Peran utama GUI adalah untuk memudahkan pengguna dalam menjalankan seluruh fitur aplikasi tanpa perlu perintah berbasis teks. Melalui GUI, pengguna dapat mengambil *screenshot* dari tampilan *video meeting*, menampilkan hasil deteksi serta pengenalan wajah, memperbarui database wajah, dan menyimpan hasil kehadiran dalam format *txt* maupun *CSV*.



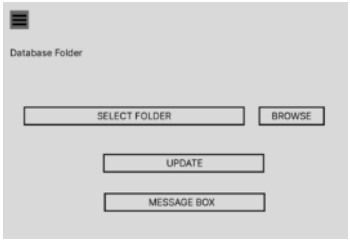
Gbr. 3 *Sketsa dasar desain tampilan utama*

Gambar 3 menampilkan tampilan utama, terdapat *main content* yang isinya berubah tergantung bagan yg dipilih. Terdapat juga tombol *burger* yang dapat menampilkan *sidebar*.

ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

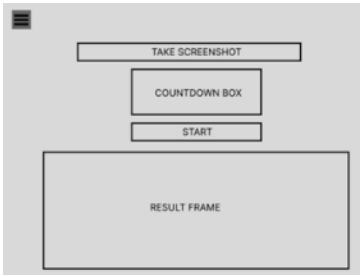
Gbr. 4 *Sketsa kasar desain sidebar*

Antarmuka utama aplikasi menampilkan jendela dengan *sidebar* navigasi di sisi kiri yang berisi tombol *Database*, *Face Recognition*, dan *Output* seperti pada gambar 4. Sidebar ini membantu pengguna berpindah antar-fungsi utama dengan mudah.



Gbr. 5 Sketsa kasar desain tampilan database

Bagian *Database* menyediakan fitur ntuk memilih folder gambar wajah dan memperbarui database seperti yang dapat dilihat pada gambar 5.



Gbr. 6 *Sketsa kasar desain tampilan face recognition*

Bagian *Face Recognition* berisi tombol *Take Screenshot* dan *Start Recognition*, yang berfungsi untuk memulai proses deteksi serta pengenalan wajah dari gambar hasil tangkapan layar. Dapat dilihat pada gambar 6.

Gbr. 7 *Sketsa kasar desain tampilan Output*

Sementara gambar 7 menunjukkan bagian *Output* menyediakan dropdown untuk memilih format hasil keluaran (*txt* atau *CSV*), tombol *Generate Output* untuk menyimpan data hasil pengenalan, serta *View Output* untuk menampilkan isi file hasil pada *message box*. Pada tampilan keluaran, hasil deteksi wajah ditampilkan dalam *Result Frame* lengkap dengan identitas peserta yang dikenali. File hasil pengenalan wajah disimpan secara otomatis di folder *image_output.*

Tata letak antarmuka dirancang dengan memperhatikan prinsip penggunaan ruang dan skala yang optimal. Setiap tombol memiliki ukuran yang proporsional agar mudah diakses, sementara jarak antar-elemen dijaga agar tampilan tidak terlihat padat. Antarmuka juga bersifat responsif terhadap perubahan resolusi layar, di mana elemen-elemen akan menyesuaikan ukuran dan posisinya secara otomatis untuk mempertahankan keteraturan visual.

Penempatan elemen didasarkan pada prioritas fungsi. Tombol utama seperti *Take Screenshot* dan *Start Recognition* ditempatkan pada area yang mudah dijangkau agar pengguna dapat menjalankan proses utama dengan cepat. Selain itu, elemen-elemen dalam GUI dikelompokkan berdasarkan kategori fungsi, yaitu *Kelola Database*, *Pengenalan Wajah*, dan *Tampilan Output*, sehingga pengguna dapat memahami hubungan antar-fitur dengan lebih mudah.

Desain GUI ini dibuat dengan pendekatan fungsional dan ergonomis untuk memberikan pengalaman interaksi yang efisien, intuitif, dan mudah dipahami. Prinsip desain tersebut memastikan bahwa seluruh komponen aplikasi dapat digunakan dengan nyaman pada berbagai ukuran layar tanpa mengorbankan kejelasan tampilan.

4) Modul Penyimpanan Data*:* Modul penyimpanan data berfungsi untuk mencatat dan menyimpan hasil proses deteksi serta pengenalan wajah yang dilakukan oleh sistem. Proses penyimpanan ini mencakup dua jenis keluaran utama, yaitu file *Comma Separated Values (CSV)* dan file teks (*TXT*), sehingga hasil pengenalan dapat diakses kembali dan dianalisis secara terstruktur.

Setiap kali proses pengenalan wajah selesai dijalankan, sistem akan menuliskan data hasil pengenalan ke dalam file keluaran secara otomatis. Data yang dicatat mencakup nama peserta, tanggal dan waktu proses pengenalan. Semua hasil penyimpanan disimpan ke dalam direktori bernama *image_output/.* Struktur folder ini berfungsi sebagai wadah utama bagi file hasil tangkapan layar dan file hasil pengenalan.

C. Validasi

Validasi dilakukan dengan menjalankan seluruh proses sistem dari awal hingga akhir. Pengguna mengambil tangkapan layar dari *video meeting*, kemudian sistem mendeteksi wajah pada gambar, mengenali identitas yang sesuai dari *database JSON*, menampilkan hasil di GUI, dan menyimpan catatan ke file keluaran.

Hasil validasi menunjukkan bahwa setiap modul bekerja sesuai fungsi dan dapat diintegrasikan dengan baik. Seluruh rancangan sistem telah sesuai dengan tujuan penelitian, yaitu menghasilkan model aplikasi pengenalan wajah yang siap untuk tahap implementasi dan pengujian performa pada penelitian berikutnya.

IV. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi sistem pengenalan wajah untuk pendataan kehadiran peserta *video meeting* berdasarkan kode program yang telah dikembangkan. Implementasi meliputi empat modul utama yaitu deteksi wajah (Haarcascade), pengenalan wajah (FaceNet dengan embedding dan jarak *Euclidean*), antarmuka pengguna (GUI) berbasis *customtkinter*, serta penyimpanan data dalam berkas teks, CSV, dan citra beranotasi. Seluruh proses bekerja berbasis tangkapan layar (*screenshot*), bukan kamera langsung.

ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

A. Hasil Rancangan Sistem

Sistem dioperasikan melalui GUI dengan tiga kelompok fungsi utama yang dapat diakses melalui *sidebar*, yaitu Database, Face Recognition, dan Output. Pada bagian Face Recognition, pengguna menekan Take Screenshot untuk mengambil tangkapan layar *video meeting*. Program melakukan *countdown*, menyembunyikan jendela aplikasi sementara, menjalankan *ImageGrab* untuk menangkap layar, lalu memulihkan jendela dan menyimpan citra ke folder *screenshots*.

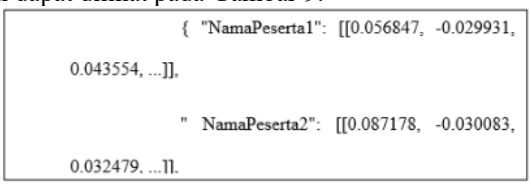
Ketika Start Recognition dijalankan, sistem memuat citra pada *screenshots*, mendeteksi wajah menggunakan Haarcascade, mengekstraksi embedding menggunakan model FaceNet, kemudian membandingkan embedding tersebut dengan data pada *face sign.json* menggunakan jarak *Euclidean*. Jika jarak minimum berada di bawah ambang batas (*threshold*) yang ditetapkan, identitas dinyatakan cocok; jika tidak, diberi label Unknown. Hasil pengenalan divisualisasikan pada GUI dan citra beranotasi disimpan ke folder *image_output*.

B. Implementasi Antarmuka Pengguna

Antarmuka pengguna dikembangkan menggunakan *customtkinter*. Tampilan Face Recognition menampilkan tombol Take Screenshot, Start Recognition, area informasi proses, serta *result frame* untuk citra hasil anotasi. Tampilan ini dapat dilihat pada Gambar 8.

Gbr. 8 *Tampilan halaman face recognition*

Menu Database digunakan untuk memperbarui basis data wajah: pengguna memilih direktori foto wajah, sistem mendeteksi area wajah, melakukan *resize* ke 160x160, mengekstraksi embedding, lalu menyimpan pasangan {nama: vektor} ke dalam *face sign.json*. contoh penyimpanan pada json ini dapat dilihat pada Gambar 9.



Gbr. 9 *Struktur face sign.json*

Menu Output menyediakan pemilihan format hasil (*txt* atau *CSV*) melalui *option menu*, tombol Generate Output untuk membentuk atau menambah berkas rekap, serta View Output File untuk menampilkan isi berkas pada *textbox*. Struktur ini memudahkan rekap kehadiran tanpa operasi terminal, sebagaimana ditunjukkan pada Gambar 10.

Gbr. 10 *Tampilan halaman output*

C. Hasil Validasi Fungsional

Validasi fungsional difokuskan pada keterhubungan modul dan konsistensi keluaran.

- Deteksi dan Ekstraksi Wajah. Fungsi *detectMultiScale()* dijalankan dengan parameter moderat (misalnya *scaleFactor = 1.1* dan *minNeighbors = 4*) baik pada tahap pembaruan database (ekstraksi embedding awal) maupun saat pengenalan di halaman Face Recognition. Hasil deteksi dikrop dan di-*resize* (160x160) sebelum diekstraksi embedding, sehingga masukan ke FaceNet konsisten.
- Pengenalan dan Keputusan. Identitas ditentukan oleh jarak *Euclidean* minimum terhadap embedding pada *face_sign.json*. Bila melebihi ambang *threshold*, sistem menetapkan *Unknown*. Mekanisme ini sederhana, transparan, dan mudah diaudit.
- Pencatatan Hasil. Selama sesi pengenalan, identitas yang dikenali dicatat ke *log* dengan format "timestamp – identity". Saat Generate Output, *log* diolah menjadi *recognition_output.txt* atau *recognition_output.csv*. Sistem menghindari duplikasi dengan memeriksa entri yang sudah ada.

Contoh format keluaran ditunjukkan pada Tabel 1. Dapat dilihat pada Tabel 1 bahwa sistem merekap waktu dan identitas pada dua kolom sederhana yang siap dipakai untuk rekap kehadiran dan analisis lanjutan.

TABEL I
OUTPUT PENYIMPANAN DATA

No	Nama	Waktu Pendataan
1	NamaPeserta1	2025-10-31 15:30:24
2	NamaPeserta2	2025-10-31 15:30:24
3	Tidak dikenal	2025-10-31 15:30:24

Selain itu, citra hasil anotasi (kotak deteksi dan label identitas) disimpan otomatis ke *image_output* dengan penamaan berbasis *timestamp* untuk memudahkan pelacakan dan audit.

D. Hasil Uji

Uji implementasi sistem dilakukan menggunakan satu tangkapan layar (screenshot) dari sesi Zoom meeting yang berisi 15 wajah peserta dengan variasi posisi, pencahayaan,

ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

dan jarak kamera. Citra tersebut diambil langsung melalui tombol Take Screenshot pada antarmuka aplikasi.

Proses pengujian dimulai dengan mendeteksi seluruh wajah pada citra tersebut menggunakan Haarcascade. Setiap area wajah yang berhasil dideteksi kemudian diekstraksi menjadi embedding vektor oleh model FaceNet, dan dibandingkan dengan basis data wajah yang tersimpan pada berkas face_sign.json. Hasil identifikasi divisualisasikan dengan kotak hijau pada area wajah dan teks label berwarna cyan di atas setiap kotak, sebagaimana ditampilkan pada Gambar 11.

Gbr. 11 *Hasil pengujian sistem terhadap satu frame Zoom berisi 15 wajah peserta*

Tabel 2 berikut menampilkan ringkasan hasil pengujian sistem terhadap citra tersebut. Berdasarkan hasil pada Tabel 2, sistem menunjukkan performa deteksi dan pengenalan yang berfungsi dengan baik pada kondisi nyata. Dari total 15 wajah dalam satu frame, 10 wajah berhasil dikenali dengan benar, 1 wajah terdeteksi namun salah label, dan 4 wajah tidak terdeteksi.

TABEL II
RINGKASAN HASIL UJI

Kategori	Jumlah	Presentase *	Keterangan
Terdeteksi & akurat	10	66,7%	label sesuai database (berdasarkan *face_sign.json*)
Terdeteksi namun tidak akurat	1	6,7%	Kemungkinan data pelatihan wajah tersebut kurang
Tidak terdeteksi	4	26,6%	Wajah tidak jelas/terpotong pada tangkapan layar
Total	15	100%	

Secara visual, hasil anotasi memperlihatkan bahwa sistem mampu mengidentifikasi mayoritas wajah frontal dengan pencahayaan cukup. Kegagalan deteksi sebagian besar disebabkan oleh posisi wajah yang tidak menghadap kamera (non-frontal), ukuran wajah terlalu kecil dalam frame, atau gangguan pencahayaan seperti backlight dan noise kamera.

Untuk menilai performa implementasi secara lebih kuantitatif, dilakukan perhitungan metrik dasar berupa accuracy, precision, recall, dan F1-score berdasarkan hasil pengujian ini. Penghitungan dilakukan pada dua tingkat, yaitu deteksi wajah dan pengenalan identitas (end-to-end). Definisi metrik yang digunakan sebagai berikut:

- True Positive (TP): wajah terdeteksi dan dikenali dengan label benar
- False Positive (FP): wajah terdeteksi namun salah identifikasi
- False Negative (FN): wajah tidak terdeteksi
- Accuracy = TP / Total wajah
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1-score = 2 × Precision × Recall / (Precision + Recall)

Berdasarkan hasil pengujian diperoleh data sebagai berikut:

- Total wajah pada frame = 15
- TP = 10 (terdeteksi dan akurat)
- FP = 1 (terdeteksi tetapi tidak akurat)
- FN = 4 (tidak terdeteksi)

Dari data tersebut diperoleh nilai metrik berikut: Precision = 10 / (10 + 1) = 0,909 atau 90,9% Recall = 10 / (10 + 4) = 0,714 atau 71,4% Accuracy = 10 / 15 = 0,667 atau 66,7% F1-score = 2 × 0,909 × 0,714 / (0,909 + 0,714) = 0,800 atau 80,0%

Selain itu, pada tingkat deteksi wajah (tanpa mempertimbangkan kesalahan label), diperoleh nilai recall deteksi sebesar 11 / (11 + 4) = 73,3%. Keterangan: tanda "–" menunjukkan nilai yang tidak dihitung dalam penelitian ini.

Baris pertama pada Tabel 3 menunjukkan performa tahap deteksi wajah oleh algoritma Haarcascade. Nilai recall 73,3% menunjukkan bahwa dari total 15 wajah pada frame, sistem berhasil mendeteksi 11 di antaranya sebagai wajah, sedangkan empat wajah tidak terdeteksi karena kondisi visual yang kurang baik.

TABEL III
METRIK PENGUJIAN PADA SATU FRAME ZOOM (15 WAJAH)

Level Evaluasi	TP	FP	FN	Total	Precision	Recall	Accuracy	F1-Score
Deteksi	11	-	4	15	-	73,3%	-	-
End-to-end	10	1	4	15	90,9%	71,4%	66,7%	80,0%

Baris kedua menunjukkan performa sistem secara end-to-end, yaitu keseluruhan proses mulai dari deteksi hingga pengenalan identitas menggunakan FaceNet. Nilai precision sebesar 90,9% menandakan bahwa ketika sistem berhasil mendeteksi wajah, label identitas yang diberikan hampir selalu benar. Nilai recall sebesar 71,4% menunjukkan bahwa masih ada sebagian wajah yang tidak berhasil terdeteksi atau dikenali. Nilai accuracy 66,7% menunjukkan proporsi wajah yang benar-benar dikenali dengan label tepat terhadap seluruh wajah pada frame, sedangkan F1-score sebesar 80,0% menunjukkan keseimbangan yang baik antara ketepatan pengenalan dan kemampuan sistem mendeteksi wajah.

Secara keseluruhan, hasil ini menunjukkan bahwa sistem telah berfungsi operasional dengan performa pengenalan yang cukup baik pada kondisi nyata, terutama ketika wajah terlihat jelas dan frontal. Namun, akurasi masih dipengaruhi oleh kualitas tangkapan layar dan keterbatasan data latih.

Penelitian ini memiliki keterbatasan pada jumlah dan variasi data pelatihan wajah yang digunakan. Basis data face_sign.json hanya berisi beberapa contoh per individu, sehingga belum dapat merepresentasikan variasi ekspresi, pencahayaan, dan sudut wajah yang berbeda. Selain itu, pengujian baru dilakukan pada satu frame tangkapan layar, sehingga hasil metrik yang diperoleh belum sepenuhnya

ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

menggambarkan performa sistem pada skenario yang lebih luas.

Untuk penelitian selanjutnya, disarankan untuk memperbanyak jumlah dataset wajah, menambah variasi kondisi pencahayaan dan posisi wajah, serta melakukan pengujian pada lebih banyak frame agar perhitungan metrik menjadi lebih representatif dan akurat.

V. KESIMPULAN

Penelitian ini berhasil mengimplementasikan sistem pendataan kehadiran berbasis pengenalan wajah menggunakan *Python* dengan penerapan model *Incremental* yang terdiri dari empat modul utama, yaitu deteksi wajah, pengenalan wajah, antarmuka pengguna (*GUI*), dan penyimpanan data. Sistem ini menggabungkan algoritma *Haarcascade* untuk deteksi wajah, *FaceNet* untuk pengenalan berbasis *CNN*, serta *Graphical User Interface* (*GUI*) berbasis customtkinter sebagai sarana interaksi yang memudahkan pengguna menjalankan seluruh fungsi tanpa perintah teks.

Hasil pengujian pada satu frame *Zoom* menunjukkan bahwa sistem telah berfungsi sesuai rancangan dengan *precision* 90,9%, *recall* 71,4%, *accuracy* 66,7%, dan *F1-score* 80,0%. Nilai tersebut menunjukkan bahwa pipeline deteksi–pengenalan bekerja stabil pada kondisi visual yang memadai. Keterbatasan penelitian ini meliputi jumlah data latih yang terbatas dan pengujian yang masih dilakukan pada satu frame. Pengembangan lebih lanjut disarankan dengan memperbanyak dataset, memperluas variasi pencahayaan dan pose wajah, serta mengoptimalkan modul deteksi dan *GUI* agar sistem dapat beroperasi secara lebih akurat dan efisien pada berbagai kondisi nyata.

UCAPAN TERIMA KASIH

Penulis berterima kasih kepada seluruh pihak yang telah berkontribusi dalam penelitian ini, khususnya Universitas Pembangunan Nasional "Veteran" Jawa Timur, serta rekan-rekan yang turut membantu dalam proses pengujian dan pengembangan sistem.

REFERENSI

[1] Z. H. Lin and Y. Z. Li, "Design and implementation of classroom attendance system based on video face recognition," in Proc. 2019 Int. Conf. Intelligent Transportation, Big Data & Smart City (ICITBS), pp. 385–388, IEEE, Jan. 2019.

[2] M. Firdaus, "Pengenalan wajah dengan algoritma Local Binary Pattern Histogram menggunakan Python," REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer, vol. 6, no. 2, pp. 272–281, 2022.

[3] F. Alassery, "A smart classroom of wireless sensor networks for students time attendance system," in Proc. 2019 IEEE Integrated STEM Education Conf. (ISEC), pp. 324–331, IEEE, Mar. 2019.

[4] S. Khan, A. Akram, and N. Usman, "Real-time automatic attendance system for face recognition using face API and OpenCV," Wireless Personal Communications, vol. 113, no. 1, pp. 469–480, 2020.

[5] H. Haqqi, "Prototipe sistem pendataan akses parkir menggunakan pengenalan wajah dengan OpenCV Python," Doctoral dissertation, Institut Teknologi Telkom Jakarta, 2022.

[6] T. Susim and C. Darujati, "Pengolahan citra untuk pengenalan wajah (Face Recognition) menggunakan OpenCV," Jurnal Syntax Admiration, vol. 2, no. 3, pp. 534–545, 2021.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. 2005 IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893, June 2005.

[8] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 1867–1874, 2014.

[9] K. Zhang, Z. Zhang, and Y. Yang, "Joint face detection and alignment using multitask cascaded convolutional networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.

[10] A. I. Warnilah, H. Sutisna, A. Jaya-Mulyana, F. Siti-Nuraeni, and T. A. Widianto, "Program aplikasi pendeteksi masker dengan menggunakan algoritma Haarcascade," EVOLUSI: Jurnal Sains dan Manajemen, vol. 10, no. 1, 2022.

[11] D. Aldiani, G. Dwilestari, H. Susana, R. Hamonangan, and D. Pratama, "Implementasi algoritma CNN dalam sistem absensi berbasis pengenalan wajah," Jurnal Informatika Polinema, vol. 10, no. 2, pp. 197–202, 2024.

[12] N. C. I. Natun, M. A. Santhia, and Y. R. Kaesmetan, "Identifikasi pengenalan wajah berdasarkan jenis kelamin menggunakan metode Convolutional Neural Network (CNN)," *Journal of Technology and Informatics (JoTI)*, vol. 6, no. 1, pp. 50–57, 2024.

[13] I. Ikmal, M. Masnur, and H. Hamra, "Absensi perpustakaan pengenalan wajah berbasis Open Computer Vision," *Jurnal Sintaks Logika*, vol. 5, no. 1, pp. 82–92, 2025.

[14] A. E. Pramudit and M. B. Akbar, "Absensi dengan pengenalan wajah menggunakan Convolutional Neural Network (CNN) dan Euclidean Distance," *Jurnal Info Digit (JID)*, vol. 2, no. 2, pp. 616–631, 2024.

[15] R. A. Firdaus, E. D. Wahyuni, and A. Agussalim, "Rancang bangun sistem presensi pegawai berbasis geo lokasi dan pengenalan wajah menggunakan FaceNet," Jurnal Media Infotama, vol. 20, no. 2, pp. 410–416, 2024.