

PENGEMBANGAN SISTEM DETEKSI WAJAH MENGUNAKAN DEEP LEARNING UNTUK APLIKASI ABSENSI OTOMATIS DI PERUSAHAAN ARIS MOTOR

Muhammad Bahrul Ulum¹

¹Program Studi Teknik Informatika STMIK DCI

email : sarbahrulum@gmail.com⁽¹⁾

Abstrak

Presensi karyawan merupakan salah satu hal penting dalam manajemen perusahaan. Namun, proses absensi manual seringkali menimbulkan kendala seperti antrian, pencatatan yang kurang akurat, dan potensi kecurangan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem absensi otomatis berbasis deteksi wajah dengan memanfaatkan teknologi Deep Learning, khususnya metode *Convolutional Neural Network (CNN)*, menggunakan bahasa pemrograman Python. Sistem ini dirancang untuk mendeteksi dan mengenali wajah karyawan secara otomatis melalui kamera, sehingga proses absensi dapat dilakukan lebih cepat, efisien, dan minim sentuhan. Proses pelatihan model dilakukan dengan mengumpulkan data wajah karyawan yang kemudian diolah menjadi embedding menggunakan CNN. Selanjutnya, hasil deteksi wajah dicocokkan dengan data yang tersimpan di dalam database. Penelitian ini dilakukan di perusahaan Aris Motor sebagai lokasi uji coba sistem. Hasil pengujian menunjukkan bahwa sistem mampu mengenali wajah karyawan dengan tingkat akurasi yang baik dan memberikan umpan balik seperti "Selamat datang [nama]" saat pertama kali absen, serta mendeteksi jika wajah telah absen sebelumnya atau tidak dikenali sama sekali. Dengan adanya sistem ini, diharapkan proses absensi di Aris Motor menjadi lebih modern, praktis, dan terpercaya, sekaligus menjadi langkah awal menuju digitalisasi sistem kehadiran di lingkungan kerja.

Kata Kunci: Absensi Otomatis, Deteksi Wajah, Deep Learning, CNN, Python, Aris Motor.

1. Pendahuluan

Di Perusahaan Aris Motor, proses presensi masih menggunakan metode konvensional seperti tanda tangan manual atau pencatatan kehadiran melalui kertas. Metode ini dinilai kurang efisien karena membutuhkan waktu yang lama, rentan terhadap kesalahan pencatatan, dan berisiko terhadap tindakan kecurangan seperti titip presensi. Semakin berkembangnya teknologi informasi, kebutuhan akan sistem presensi yang otomatis, akurat, dan andal menjadi sangat penting. Salah satu teknologi yang potensial untuk mengatasi permasalahan tersebut adalah sistem deteksi wajah berbasis deep learning. Sistem ini memungkinkan proses identifikasi karyawan hanya dengan memindai wajah menggunakan kamera, tanpa perlu menyentuh alat atau melakukan input manual. Teknologi ini sangat relevan di era industri modern, terutama dalam mendukung efisiensi operasional dan keamanan data kehadiran. Di Perusahaan Aris Motor, penerapan sistem absensi berbasis deteksi wajah diharapkan mampu meningkatkan efisiensi kerja, meminimalkan potensi kecurangan absensi, serta mempermudah manajemen dalam mengelola data kehadiran secara digital dan terintegrasi. Dengan demikian, pengembangan sistem ini menjadi langkah strategis dalam mendorong transformasi digital dan modernisasi proses bisnis di perusahaan tersebut. Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk merancang dan mengembangkan sistem absensi otomatis berbasis deteksi wajah menggunakan metode deep learning, serta menguji keefektifannya dalam meningkatkan akurasi dan efisiensi manajemen kehadiran di Perusahaan Aris Motor.

2. Landasan Teori

2.1 Sistem Presensi

Sistem Presensi adalah suatu sistem yang digunakan untuk mencatat kehadiran seseorang, khususnya dalam konteks ini merupakan kehadiran pegawai. Presensi sangat berpengaruh dalam penggajian karyawan hal ini disebabkan dokumentasi kehadiran karyawan tercatat dalam sistem presensi. Sehingga saat ini untuk membantu kinerja dari admin dalam menginput data absensi karyawan banyak sistem presensi yang sudah menggunakan aplikasi seperti sistem presensi menggunakan pengenalan wajah, RFID, barcode, dan lain-lain. (Zihamussholihin dkk, 2021).

2.2 Sistem Presensi Otomatis

Sistem presensi otomatis merupakan sistem pencatatan kehadiran yang menggunakan teknologi untuk mencatat dan memverifikasi kehadiran secara otomatis. Teknologi yang digunakan dapat berupa RFID, biometrik, QR code, hingga pengenalan wajah, (Erlinda, 2019).

Yuliana dan Nugroho (2019) menyatakan bahwa sistem presensi berbasis pengenalan wajah mampu meningkatkan kedisiplinan serta efisiensi pencatatan kehadiran di lingkungan kerja.

2.3 Pengenalan Wajah

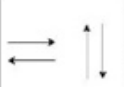



Pengenalan Wajah atau Face Recognition merupakan pengembangan dari teknologi deteksi wajah dimana teknologi ini dapat menghasilkan wajah dari hasil tangkapan kamera dan melakukan deteksi persamaan wajah dengan data wajah yang diketahui komputer, sehingga komputer dapat mengenali dan atau mengetahui keberadaan seseorang (Fajri dkk, 2020).









2.4 Face Recognition

Face Recognition adalah teknologi komputer untuk mengidentifikasi atau mendeteksi wajah manusia melalui sebuah gambar digital dengan cara mencocokkan tekstur lekuk wajah manusia dengan data yang tersimpan di database (Ramadhan, 2022). Menurut Li & Jain (2019), pengenalan wajah sangat bergantung pada karakteristik unik wajah seperti jarak antar mata, bentuk rahang, dan kontur wajah.

2.5 Flow Chart









Menurut Shelly Cashman dalam buku "*Systems Analysis and Design*", flowchart membantu analis sistem dan pengembang dalam memahami bagaimana data mengalir melalui sistem serta proses-proses yang dilakukan terhadap data tersebut (Shelly, Cashman, & Rosenblatt, 2011).

| | |
|---|---|
|  | Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line. |
|  | On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama. |
|  | Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan. |
|  | Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer. |

| | | | |
|---|---|---|--|
|  | Off-Page Reference Simbol untuk keluar - masuk atau penambungan proses dalam lembar kerja yang berbeda. |  | Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak. |
|  | Terminator Simbol yang menyatakan awal atau akhir suatu program. |  | Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur. |
|  | Process Simbol yang menyatakan suatu proses yang dilakukan komputer. |  | Display Simbol yang menyatakan peralatan output yang digunakan. |
|  | Decision Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak. |  | Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal. |

2.6 Flow Map

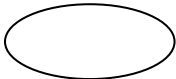
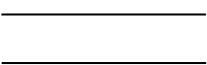
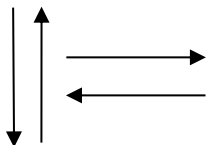
Menurut Kendall & Kendall (2011), flow diagram berguna untuk mendeskripsikan bagaimana informasi atau data mengalir dalam suatu sistem dan bagaimana informasi tersebut diproses, dikendalikan, dan disimpan. Flowmap biasanya digunakan pada tahap awal analisis sistem untuk membantu menggambarkan proses secara global.

| No | Gambar | Nama | Keterangan |
|----|---|-------------------------|--|
| 1 |  | Dokumen | Menunjukkan dokumen <i>input/output</i> baik untuk proses manual, mekanik atau komputer. |
| 2 |  | Kegiatan manual | Menunjukkan pekerjaan manual |
| 3 |  | Simpanan <i>offline</i> | <i>File</i> komputer yang diarsip urut |
| 5 |  | Proses | Menunjukkan kegiatan proses dari operasi program komputer |
| 11 |  | Keyboard | Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i> |
| 14 |  | Garis Alir | Menunjukkan arus dari proses |
| 15 |  | keputusan | Keputusan dalam suatu program |
| 16 |  | Display | Menampilkan ke monitor |

Pengembangan Sistem Deteksi Wajah Menggunakan Deep Learning Untuk Aplikasi Absensi Otomatis Di Perusahaan Aris Motor



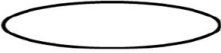
2.7 DFD


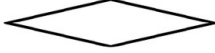

Menurut Hartono (2005:700) didalam bukunya Analisis dan desain Sistem Informasi mendefinisikan Data Flow Diagram (DFD) merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*Structured Analysis and Design*).

| Simbol | Keterangan |
|---|---|
|  | Merupakan simbol proses yang berfungsi untuk mentransfortasikan data secara umum. |
|  | Merupakan berkas atau tempat menyimpan data atau file. |
|  | Merupakan aliran data yang menggambarkan aliran data dari suatu proses ke proses yang lain. |

2.8 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) merupakan salah satu alat bantu dalam pemodelan data yang digunakan untuk menggambarkan struktur logika dari suatu basis data secara konseptual. ERD pertama kali diperkenalkan oleh Peter Chen pada tahun 1976 sebagai metode untuk merepresentasikan hubungan antar data dalam sistem basis data relasional.

| Simbol | Deskripsi |
|---|---|
| Entitas  | Data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| Atribut  | Field atau kolom data yang butuh disimpan dalam suatu entitas. |
| Atribut kunci  | Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer bersifat unik. |

| | |
|---|--|
| <p>Atribut multivali/multivalue</p>  | <p>Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p> |
| <p>Relasi</p>  | <p>Merupakan penghubung antar entitas; biasanya diawali dengan kata kerja.</p> |
| <p>Konektor</p>  | <p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakai; kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan yang lainnya (kardinalitas).</p> |

2.9 Numpy

Menurut Sardi et al. (2021:98) NumPy merupakan modul yang sangat berguna dalam analisis data dan pemodelan matematika. NumPy memiliki peran utama dalam melakukan operasi pada data numerik dalam bentuk array, seperti operasi aritmatika, trigonometri, fungsi matematika, serta statistik.

Menurut Journal of Student Development Information System (2022:123), NumPy adalah library yang digunakan untuk komputasi numerik dalam Python. NumPy menyediakan array multidimensi yang efisien dan operasi matematika yang cepat.

2.10 OpenCV

Menurut Jusia et al. (2021:34), OpenCV adalah pustaka open-source untuk pemrosesan dan analisis gambar secara real-time yang mudah diintegrasikan dengan Python. OpenCV awalnya dirilis oleh Intel pada tahun 2000 dan kini menjadi *library* standar dalam aplikasi visi komputer, seperti deteksi wajah, pelacakan objek, dan *augmented reality*.

2.11 MySQL

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) yang bersifat *open-source* dan banyak digunakan dalam pengembangan aplikasi berbasis web. *MySQL* terkenal karena performanya yang cepat, keandalan yang tinggi, serta dukungan terhadap berbagai platform dan sistem operasi. Dalam dunia akademik

maupun industri, *MySQL* sering digunakan karena mampu menangani data dalam jumlah besar, serta mudah diintegrasikan dengan berbagai bahasa pemrograman seperti PHP, Python, dan Java (Wahana Komputer, 2017; Tim IT Guna Pena, 2022).

2.12PhpMyAdmin

Menurut Standsyah dan Restu (2017: 39), *PhpMyAdmin* adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi database MySQL melalui jaringan lokal maupun internet. *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan Politeknik Negeri Sriwijaya 19 BAB II Tinjauan Pustaka (*permissions*), dan lain-lain)". Menurut Arief dalam Firliana dkk (2020: 46), *phpMyAdmin* adalah salah satu aplikasi GUI (*Graphic User Interface*) yang digunakan untuk mengelola database MySQL". Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *PhpMyAdmin* adalah sebuah aplikasi yang menangani administrasi database *MySQL*.

2.13Python

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. Dalam beberapa tahun terakhir, Python menjadi salah satu bahasa pemrograman yang sangat populer. Penulisan kode program Python dapat dilakukan melalui berbagai IDE seperti Visual Studio Code, Sublime Text, dan PyCharm, maupun melalui platform daring seperti Jupyter Notebook dan Google Colab (Riziq Sirfatullah Alfarizi dkk., 2023).

2.14Visual Studio Code

Visual Studi Code ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem oprasi *multiplatform*, artinya tersedia juga untuk versis selain windows. Visual Studio Code Juga merupakan sebuah text editor yang sangat ringan, namun menyediakan beragam fitur canggih seperti penyorortan sintaks, auto-completion, debugging internal dan integritas ke

Git, dalam konteks pengembangan perangkat lunak di Indonesia, Jurnal Teknik Indonesia (2023).

3. Analisis Masalah

Sistem presensi yang digunakan Aris Motor saat ini, kemungkinan masih manual atau mengandalkan metode konvensional, menimbulkan berbagai permasalahan yang signifikan. Permasalahan utama meliputi ketidakakuratan data absensi akibat kesalahan manusia atau potensi manipulasi, serta risiko kecurangan seperti "titip absen" yang sulit dideteksi. Selain itu, proses rekapitulasi absensi secara manual sangat tidak efisien, membuang banyak waktu staf administrasi dan berpotensi menimbulkan antrean panjang saat jam masuk kerja, yang pada akhirnya mengurangi produktivitas karyawan dan menambah beban biaya operasional perusahaan.

Dampak dari permasalahan ini sangat nyata, mulai dari kerugian finansial akibat pembayaran gaji yang tidak sesuai, penurunan disiplin karyawan, hingga penghambatan dalam pengambilan keputusan oleh manajemen karena sulitnya akses data absensi yang akurat dan *real-time*.

3.1 Analisis Kebutuhan Sistem

Dalam merancang sebuah sistem informasi yang efektif dan efisien, penting untuk terlebih dahulu memahami kebutuhan dari sistem yang akan dibangun. Analisis kebutuhan sistem bertujuan untuk mengidentifikasi fungsi-fungsi apa saja yang harus dimiliki oleh sistem agar mampu menyelesaikan permasalahan yang ada serta memenuhi harapan pengguna.

Sistem absensi otomatis berbasis deteksi wajah ini dirancang untuk menggantikan metode absensi manual yang masih digunakan di Perusahaan Aris Motor. Oleh karena itu, sistem harus mampu menangani seluruh proses mulai dari pendaftaran wajah karyawan, pengenalan wajah secara otomatis saat karyawan hadir, hingga pencatatan data kehadiran secara akurat dan *real-time*.

3.2 Kebutuhan Fungsional

Kebutuhan Fungsional merupakan kebutuhan yang harus dimiliki oleh sistem untuk dapat berjalan sebagaimana mestinya. Berikut adalah beberapa kebutuhan fungsional sistem absensi berbasis face-recognition :

- 1) Sistem harus menangkap wajah secara *real-time* melalui kamera.

Sistem akan mengaktifkan kamera setiap kali proses absensi dimulai, lalu mendeteksi wajah yang masuk ke dalam *frame*.

- 2) Sistem harus mampu mendeteksi dan mengenali wajah pengguna yang telah terdaftar.

Sistem akan menggunakan model CNN untuk mengekstraksi ciri khas wajah dan mencocokkannya dengan data embedding yang tersimpan di database/ dalam folder dataset.

- 3) Sistem harus menyimpan data absensi ke dalam basis data.

Jika wajah dikenali, maka data kehadiran seperti ID karyawan, nama waktu absen dicatat secara otomatis.

- 4) System akan menampilkan notifikasi kepada pengguna.

Pesan dalam frame camera seperti "Selamat Datang [nama]", lalu pesan dalam terminal yang menyatakan "Sudah Absen hari ini", atau informasi "Wajah Tidak Dikenali". Akan muncul sebagai umpan balik.

3.3 Sumber Dataset

Dataset wajah dikumpulkan secara mandiri dengan cara mengambil foto wajah dari seluruh karyawan perusahaan Aris Motor. Proses pengumpulan dilakukan secara langsung menggunakan kamera laptop/PC yang terintegrasi dengan sistem, atau melalui proses registrasi wajah pada sistem absensi. Setiap karyawan diminta menghadap kamera dalam berbagai ekspresi dan sudut pandang untuk meningkatkan akurasi model dalam proses pelatihan dan pengenalan wajah.

- 1) Jumlah subjek (karyawan) : \pm [jumlah karyawan] orang
- 2) Jumlah total gambar per individu : \pm [misal: 20–30 gambar per orang]
- 3) Variasi data : Pencahayaan, ekspresi wajah (senyum, netral), sudut (miring kiri, kanan, depan)
- 4) Format dan resolusi gambar: JPEG/PNG, resolusi 640x480 px

Keunggulan Pengumpulan Data Sendiri

- 1) Data lebih relevan dan spesifik terhadap individu yang akan dikenali oleh sistem.
- 2) Memberikan hasil pelatihan model yang lebih akurat karena tidak ada data wajah orang lain yang tidak relevan.
- 3) Menghindari isu privasi dan etika penggunaan dataset publik.

3.4 Karakteristik Dataset mpak

Dataset wajah yang digunakan dalam pengembangan sistem ini memiliki karakteristik sebagai berikut:

1) Jumlah Data

- Jumlah subjek (karyawan): \pm [misalnya: 25 orang]
- Jumlah gambar per individu: \pm 20–30 gambar
- Total dataset: Sekitar 500–750 gambar

Dataset ini dikumpulkan secara mandiri dari seluruh karyawan perusahaan Aris Motor melalui proses registrasi wajah menggunakan kamera langsung dari sistem.

2) Variasi Wajah

- Ekspresi wajah : Netral, Tersenyum, Sedikit berbicara atau bergerak
- Sudut pengambilan gambar : Frontal (depan), Sedikit miring ke kiri dan ke kanan, Atas dan bawah terbatas
- Aksesoris : Beberapa gambar diambil dengan/ tanpa kacamata, Rambut tertutup sebagian (misalnya jilbab bagi wanita)

3) Kondisi Pencahayaan

Beragam kondisi pencahayaan, untuk meningkatkan robust-nya sistem:

- Pencahayaan alami (siang hari)
- Pencahayaan lampu ruangan (putih/kuning)
- Pencahayaan rendah (redup)

Pencahayaan tidak dikontrol secara ketat untuk mencerminkan kondisi nyata penggunaan sistem di kantor.

4) Format dan Resolusi

- Format file : JPG dan PNG
- Resolusi : Rata-rata 640x480 px
- Gambar disimpan dalam folder terpisah berdasarkan nama karyawan sebagai label.

5) Tujuan Pengumpulan

- Digunakan untuk pelatihan model CNN agar mampu mengenali wajah masing-masing karyawan dengan akurasi tinggi.
- Dataset dibagi menjadi data latih dan data uji untuk proses evaluasi model:

Pengembangan Sistem Deteksi Wajah Menggunakan Deep Learning Untuk Aplikasi Absensi Otomatis Di Perusahaan Aris Motor

Data latih: $\pm 80\%$

Data uji: $\pm 20\%$

| No. | Jarak (Distance) | Status Pengenalan | Keterangan |
|-----|------------------|-------------------|------------|
| 1 | 0.2843 | Dikenali | Cocok |
| 2 | 0.3136 | Dikenali | Cocok |
| 3 | 0.3128 | Dikenali | Cocok |
| 4 | 0.2935 | Dikenali | Cocok |
| 5 | 0.2949 | Dikenali | Cocok |
| 6 | 0.2890 | Dikenali | Cocok |
| 7 | 0.2890 | Dikenali | Cocok |
| 8 | 0.2905 | Dikenali | Cocok |
| 9 | 0.2969 | Dikenali | Cocok |
| 10 | 0.2626 | Dikenali | Cocok |
| 11 | 0.4262 | Dikenali | Cocok |
| 12 | 0.2828 | Dikenali | Cocok |
| 13 | 0.2447 | Dikenali | Cocok |
| 14 | 0.2796 | Dikenali | Cocok |
| 15 | 0.2999 | Dikenali | Cocok |
| 16 | 0.3680 | Dikenali | Cocok |
| 17 | 0.4727 | Dikenali | Cocok |
| 18 | 0.4179 | Dikenali | Cocok |
| 19 | 0.2850 | Dikenali | Cocok |
| 20 | 0.3413 | Dikenali | Cocok |
| 21 | 0.2469 | Dikenali | Cocok |
| 22 | 0.2821 | Dikenali | Cocok |
| 23 | 0.2512 | Dikenali | Cocok |
| 24 | 0.2641 | Dikenali | Cocok |
| 25 | 0.3350 | Dikenali | Cocok |
| 26 | 0.2033 | Dikenali | Cocok |
| 27 | 0.1859 | Dikenali | Cocok |
| 28 | 0.1990 | Dikenali | Cocok |
| 29 | 0.1528 | Dikenali | Cocok |
| 30 | 0.2200 | Dikenali | Cocok |
| 31 | 0.1527 | Dikenali | Cocok |
| 32 | 0.1744 | Dikenali | Cocok |
| 33 | 0.1479 | Dikenali | Cocok |

| | | | |
|----|--------|----------|-------|
| 34 | 0.2008 | Dikenali | Cocok |
| 35 | 0.1637 | Dikenali | Cocok |
| 36 | 0.1798 | Dikenali | Cocok |
| 37 | 0.1639 | Dikenali | Cocok |
| 38 | 0.1961 | Dikenali | Cocok |
| 39 | 0.1815 | Dikenali | Cocok |
| 40 | 0.1552 | Dikenali | Cocok |

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=== MENU UTAMA ===
1. Daftarkan wajah karyawan
2. Mulai absensi
3. Lihat daftar absensi
4. Keluar
Pilih menu (1/2/3/4): 1
Masukkan nama karyawan: Bahrul Ulum
Arahkan wajah ke kamera. Kamera akan mengambil 100 gambar.
Embedding wajah "Bahrul Ulum" disimpan di: e:\Project-01\Present_face_CNN_2\embeddings\6_Bahrul Ulum.npy

=== MENU UTAMA ===
1. Daftarkan wajah karyawan
2. Mulai absensi

```

Gambar 5.1 Pendaftaran Wajah *Real-Time*

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=== MENU UTAMA ===
1. Daftarkan wajah karyawan
2. Mulai absensi
3. Lihat daftar absensi
4. Keluar
Pilih menu (1/2/3/4): 2
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.2033
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.2033 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1859
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1859 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1990
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1990 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1528

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.2033
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.2033 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1859
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1859 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1990
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1990 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1528
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1528 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.2200
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.2200 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1527
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1527 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1744

```

Gambar 5.2 Hasil Threshold Data Embedding

Pengembangan Sistem Deteksi Wajah Menggunakan Deep Learning Untuk Aplikasi Absensi Otomatis Di Perusahaan Aris Motor

Dengan threshold 0.60, 38 sistem berhasil mengenali semua wajah dengan hasil jarak yang konsisten di bawah ambang batas, menandakan kualitas embedding dan data latih yang baik.

Hasil ini menunjukkan bahwa sistem memiliki tingkat akurasi yang cukup tinggi dalam mengenali wajah karyawan dengan threshold 0.6. Penggunaan nilai ambang batas ini dinilai optimal karena mampu membedakan wajah yang serupa namun bukan orang yang sama, serta menghindari kesalahan identifikasi yang dapat mengakibatkan absensi palsu. Meski demikian, terdapat peluang peningkatan akurasi lebih lanjut dengan penyesuaian parameter lain, seperti pencahayaan saat pengambilan gambar atau peningkatan kualitas kamera.

- 1) Kinerja Sistem Pada Kondisi Pencahayaan
- 2) Pada Cahaya Terang

Nilai Embedding yang diambil

A1 : [-0.23, -0.03, 0.03, 0.08, 0.10]

A2 : [-0.25, 0.00, 0.06, -0.05, -0.00]

Menghitung Jarak Euclidiean

$$distance = \sqrt{((-0.23 + 0.25)^2 + (-0.03 - 0.00)^2 + (0.03 - 0.06)^2 + (0.08 + 0.05)^2 + (0.10 + 0.00)^2)}$$

$$distance = \sqrt{(0.0004 + 0.0009 + 0.0009 + 0.0169 + 0.01)}$$

$$distance = \sqrt{0.0291} = 0.17$$

Hasil perhitungan embedding wajah antara A1 dan A2 (yang telah dibulatkan ke dua sistem) menghasilkan jarak Euclidean sebesar 0.17, masih berada di bawah ambang batas threshold 0.6.

Maka, sistem berhasil mengenali kedua gambar sebagai orang yang sama meskipun terdapat perbedaan nilai embedding, kemungkinan karena perbedaan sudut wajah atau ekspresi. Sistem tetap bekerja dengan akurasi yang baik dalam kondisi terang.

- 1) Pada Kondisi Cahaya Redup
-

A1 : [-0.23, -0.03, 0.03, 0.08, 0.10]

A2 : [-0.25, 0.05, -0.07, 0.01, -0.15]

Menghitung Jarak Euclidiean

$$\mathit{distance} = \sqrt{((-0.23 + 0.25)^2 + (-0.03 - 0.05)^2 + (0.03 + 0.07)^2 + (0.08 + 0.01)^2 + (0.10 + 0.15)^2)}$$

Uji coba dilakukan untuk membandingkan gambar Bahrul-2 (kondisi terang) dengan Bahrul-9 (kondisi redup atau berbeda sudut). Setelah pembulatan embedding ke dua sistem dan perhitungan jarak Euclidean, diperoleh nilai 0.29, yang masih berada di bawah ambang batas threshold 0.6.

Tabel 5.2 Hasil Uji Coba Sistem

| No | Gambar 1 | Gambar 2 | Kondisi | Embedding (dibulatkan) | Jarak Euclidean | Hasil |
|----|---------------|-------------------|----------------------------|---|-----------------|------------------------|
| 1 | Gambar-1.jpeg | Gambar-2.jpeg | Sama-sama terang | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [-0.23, -0.03, 0.03, 0.08, 0.10] | 0.00 | ✔ Sama (Terdeteksi) |
| 2 | Gambar-3.jpeg | Gambar-4.jpeg | Sama-sama redup | [-0.26, 0.05, 0.01, -0.11, 0.06] vs [-0.22, 0.02, -0.06, 0.08, -0.06] | 0.24 | ✔ Sama (Terdeteksi) |
| 3 | Gambar-2.jpeg | Gambar-4.jpeg | Terang vs redup | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [-0.22, 0.02, -0.06, 0.08, -0.06] | 0.19 | ✔ Sama (Terdeteksi) |
| 4 | Gambar-2.jpeg | Gambar-5.jpeg | Terang vs redup/sudut lain | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [-0.25, 0.05, -0.07, 0.01, -0.15] | 0.29 | ✔ Sama (Terdeteksi) |
| 5 | Gambar-2.jpeg | Gambar Internet 1 | Terang vs wajah asing | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [0.21, -0.33, -0.21, -0.06, 0.02] | 0.61 | ✘ Tidak Sama (Ditolak) |
| 6 | Gambar-2.jpeg | Gambar Internet 2 | Terang vs wajah asing | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [0.45, 0.19, 0.00, -0.11, 0.05] | 0.74 | ✘ Tidak Sama (Ditolak) |
| 7 | Gambar-2.jpeg | Gambar Internet 3 | Terang vs wajah asing | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [0.32, -0.16, 0.29, -0.03, -0.13] | 0.67 | ✘ Tidak Sama (Ditolak) |

3.5 Pemilihan Arsitektur CNN yang Paling Sesuai

Berdasarkan kebutuhan sistem absensi otomatis di perusahaan Aris Motor yang menekankan pada kecepatan proses, akurasi pengenalan, dan kemudahan implementasi, maka dipilihlah kombinasi antara:

- 1) OpenCV DNN Face Detector (untuk mendeteksi wajah secara real-time dari kamera),
- 2) Dan model CNN pre-trained dari pustaka `face_recognition` (berbasis Dlib CNN) untuk menghasilkan face embedding.
Alasan pemilihan arsitektur tersebut adalah:
- 3) Efisien secara komputasi, sehingga dapat dijalankan pada perangkat standar tanpa GPU khusus.
- 4) Akurasi pengenalan wajah cukup tinggi, dengan threshold distance yang dapat disesuaikan.
- 5) Integrasi yang mudah dengan Python dan OpenCV, serta dokumentasi yang luas.

Pemilihan ini juga mempertimbangkan bahwa sistem tidak membutuhkan pelatihan ulang (retraining) model CNN dari nol, karena model pre-trained sudah mampu mengenali wajah baru melalui penambahan data embedding.

4. Perancangan Sistem

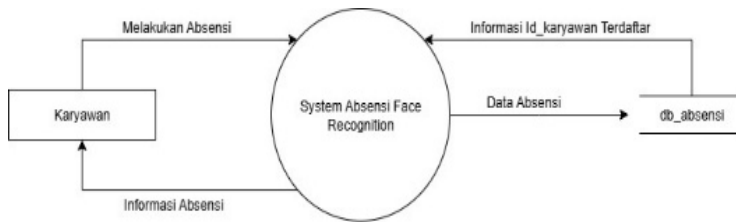
4.1 Perancangan Umum

Proses absensi manual seringkali menimbulkan kendala seperti antrian, pencatatan yang kurang akurat, dan potensi kecurangan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem absensi otomatis berbasis deteksi wajah dengan memanfaatkan teknologi Deep Learning, khususnya metode *Convolutional Neural Network (CNN)*, menggunakan bahasa pemrograman Python. Sistem ini dirancang untuk mendeteksi dan mengenali wajah karyawan secara otomatis melalui kamera, sehingga proses absensi dapat dilakukan lebih cepat, efisien, dan minim sentuhan. Proses pelatihan model dilakukan dengan mengumpulkan data wajah karyawan yang kemudian diolah menjadi embedding menggunakan CNN. Selanjutnya, hasil deteksi wajah dicocokkan dengan data yang tersimpan di dalam database

4.2 Data Flow Diagram

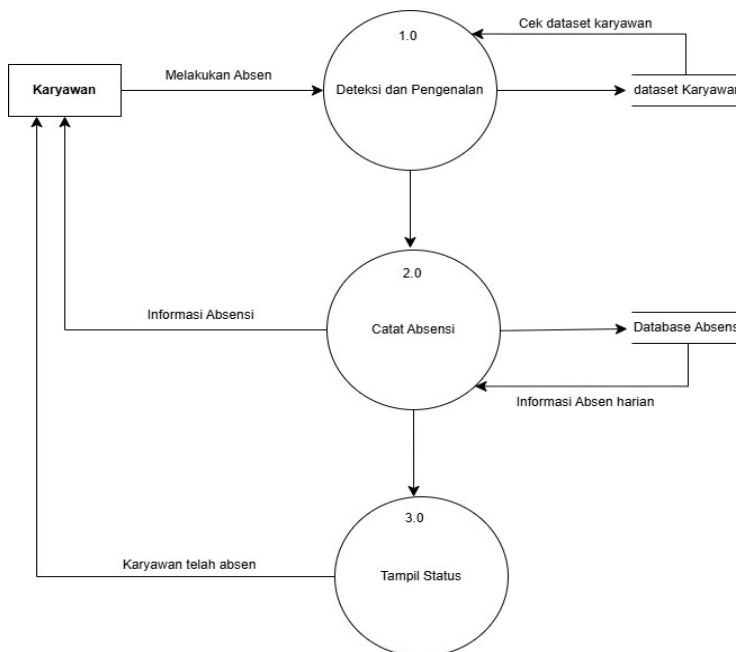
Berikut Data Flow Diagram Sistem:

- 1) Diagram Konteks :



Penjelasan : Gambar 4.1 menjelaskan bahwa karyawan dapat melakukan absensi pada system, system akan memproses data karyawan yang sebelumnya sudah tersimpan dan terdaftar pada database, jika dinyatakan cocok maka system akan memberikan informasi absen kepada karyawan.

2) Diagram Level 1:

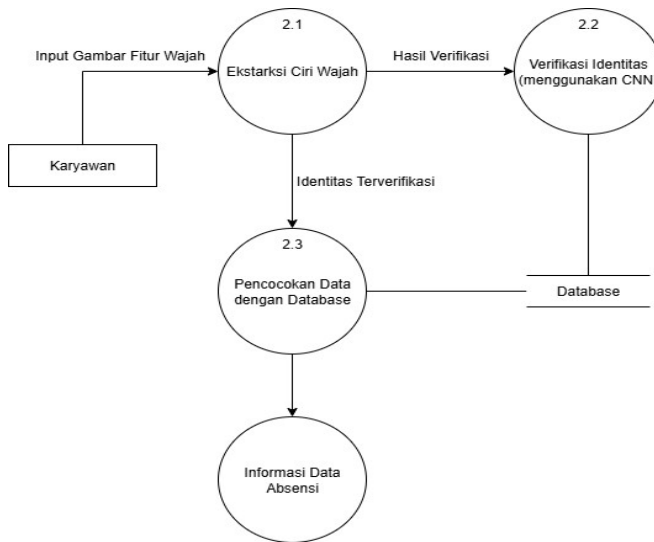


Penjelasan : pada gambar 4.2 menjelaskan bahwa karyawan melakukan absen lalu system akan mendeteksi wajah karyawan dan melakukan pengenalan wajah dari data set yang telah tersimpan, lalu system akan

Pengembangan Sistem Deteksi Wajah Menggunakan Deep Learning Untuk Aplikasi Absensi Otomatis Di Perusahaan Aris Motor

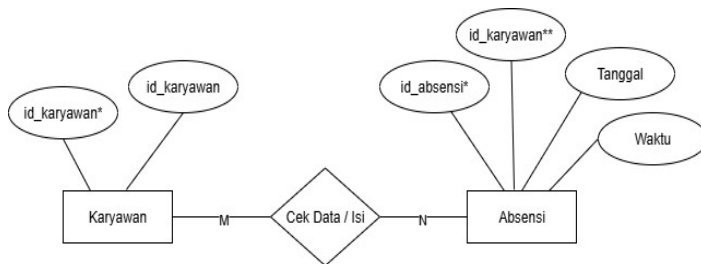
melakukan pencatatan absensi dan di simpen ke database setelahnya sistem akan menampilkan informasi absensi kepada karyawan.

3) Diagram Level 2:



Penjelasan : Dari gambar 4.3 dapat dijelaskan saat karyawan menginputkan gambar wajah system akan meng ekstrasi hasil rekam wajah karyawan, lalu memverifikasi identitas karyawan menggunakan CNN apakah terdaftar atau tidak nya, Lalu system akan melakukan pencocokan data karyawan mulai dari id nama dan fitur wajah karyawan, setelahnya system akan memberikan informasi absensi kepada karyawan.

4.3 Entity Relation Diagram (ERD)



Penjelasan : Dari gambar 4.4 menjelaskan Beberapa karyawan dapat melakukan satu kali absensi pada untuk satu hari.

5. Implementasi

Setelah melakukan perancangan maka tahap selanjutnya ialah Implementasi Sistem yang mana merupakan proses akhir dari penerapan system yang dirancang. Dimana tahap ini merupakan tahap meletakkan system agar siap untuk dioperasikan dan dipandang sebagai usaha mewujudkan system yang telah dirancang.

5.1 Kebutuhan Perangkat Keras

Adapun perangkat keras yang digunakan untuk membangun perangkat ini adalah sebagai berikut:

- 1) Kamera /Webcam.
- 2) Laptop.
- 3) Processor intel i5.
- 4) Monitor.
- 5) Keyboard.
- 6) Mouse.

5.2 Kebutuhan Perangkat Lunak

Adapun perangkat keras yang digunakan untuk membangun perangkat ini adalah sebagai berikut:

- 1) Windows 10.
- 2) Python 3.x.
- 3) Library face-recognition, openCV, dlib.
- 4) Visual Studio Code.
- 5) Conda .
- 6) MySQL/ MariaDB.

5.3 Hasil Uji Akurasi Sistem

Pengujian akurasi sistem absensi otomatis berbasis face recognition dilakukan dengan menetapkan nilai threshold sebesar 0.6 sebagai batas toleransi kecocokan data wajah. Nilai threshold ini berarti bahwa apabila jarak (distance) antara face embedding wajah yang terdeteksi dan data wajah yang

Pengembangan Sistem Deteksi Wajah Menggunakan Deep Learning Untuk Aplikasi Absensi Otomatis Di Perusahaan Aris Motor

tersimpan dalam database kurang dari atau sama dengan 0.6, maka sistem akan menganggap wajah tersebut cocok atau dikenali.

Tabel 5.1 Hasil Pengujian Face Recognition Threshold 0.6

| No. | Jarak (Distance) | Status Pengenalan | Keterangan |
|-----|------------------|-------------------|------------|
| 1 | 0.2843 | ✓ Dikenali | Cocok |
| 2 | 0.3136 | ✓ Dikenali | Cocok |
| 3 | 0.3128 | ✓ Dikenali | Cocok |
| 4 | 0.2935 | ✓ Dikenali | Cocok |
| 5 | 0.2949 | ✓ Dikenali | Cocok |
| 6 | 0.2890 | ✓ Dikenali | Cocok |
| 7 | 0.2890 | ✓ Dikenali | Cocok |
| 8 | 0.2905 | ✓ Dikenali | Cocok |
| 9 | 0.2969 | ✓ Dikenali | Cocok |
| 10 | 0.2626 | ✓ Dikenali | Cocok |
| 11 | 0.4262 | ✓ Dikenali | Cocok |
| 12 | 0.2828 | ✓ Dikenali | Cocok |
| 13 | 0.2447 | ✓ Dikenali | Cocok |
| 14 | 0.2796 | ✓ Dikenali | Cocok |
| 15 | 0.2999 | ✓ Dikenali | Cocok |
| 16 | 0.3680 | ✓ Dikenali | Cocok |
| 17 | 0.4727 | ✓ Dikenali | Cocok |
| 18 | 0.4179 | ✓ Dikenali | Cocok |
| 19 | 0.2850 | ✓ Dikenali | Cocok |
| 20 | 0.3413 | ✓ Dikenali | Cocok |
| 21 | 0.2469 | ✓ Dikenali | Cocok |
| 22 | 0.2821 | ✓ Dikenali | Cocok |
| 23 | 0.2512 | ✓ Dikenali | Cocok |
| 24 | 0.2641 | ✓ Dikenali | Cocok |

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=== MENU UTAMA ===
1. Daftarkan wajah karyawan
2. Mulai absensi
3. Lihat daftar absensi
4. Keluar
Pilih menu (1/2/3/4): 1
Masukkan nama karyawan: Bahrul UJum
📷 Arahkan wajah ke kamera. Kamera akan mengambil 100 gambar.
✓ Embedding wajah 'Bahrul UJum' disimpan di: e:\Project-01\Present_face_CNN_2\embeddings\6_Bahrul UJum.npy

=== MENU UTAMA ===
1. Daftarkan wajah karyawan
2. Mulai absensi
    
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=== MENU UTAMA ===
1. Daftarkan wajah karyawan
2. Mulai absensi
3. Lihat daftar absensi
4. Keluar
Pilih menu (1/2/3/4): 2
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.2033
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.2033 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1859
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1859 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1990
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1990 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1528
Ln 5, Col 1 Spaces:4 UTF-8 CRLF () Py

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.2033
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.2033 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1859
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1859 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1990
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1990 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1528
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1528 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.2200
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.2200 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1527
[INFO] Threshold: 0.60 - Distance Ditemukan: 0.1527 - Dikenali sebagai: Bahrul Ulum
[DEBUG] Bandingkan dengan Bahrul Ulum: distance = 0.1744
Ln 5, Col 1 Spaces:4 UTF-8 CRLF () Py

```

5.4 Kinerja Sistem Pada Kondisi Pencahayaan

| No | Gambar 1 | Gambar 2 | Kondisi | Embedding (dibulatkan) | Jarak Euclidean | Hasil |
|----|---------------|-------------------|----------------------------|---|-----------------|------------------------|
| 1 | Gambar-1.jpeg | Gambar-2.jpeg | Sama-sama terang | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [-0.23, -0.03, 0.03, 0.08, 0.10] | 0.00 | ✅ Sama (Terdeteksi) |
| 2 | Gambar-3.jpeg | Gambar-4.jpeg | Sama-sama redup | [-0.26, 0.05, 0.01, -0.11, 0.06] vs [-0.22, 0.02, -0.06, 0.08, -0.06] | 0.24 | ✅ Sama (Terdeteksi) |
| 3 | Gambar-2.jpeg | Gambar-4.jpeg | Terang vs redup | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [-0.22, 0.02, -0.06, 0.08, -0.06] | 0.19 | ✅ Sama (Terdeteksi) |
| 4 | Gambar-2.jpeg | Gambar-5.jpeg | Terang vs redup/sudut lain | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [-0.25, 0.05, -0.07, 0.01, -0.15] | 0.29 | ✅ Sama (Terdeteksi) |
| 5 | Gambar-2.jpeg | Gambar Internet 1 | Terang vs wajah asing | [-0.23, -0.03, 0.03, 0.08, 0.10] vs [0.21, -0.33, - | 0.61 | ❌ Tidak Sama (Ditolak) |

Berdasarkan hasil penelitian dan pengembangan sistem absensi otomatis berbasis pengenalan wajah (face recognition) menggunakan metode Convolutional Neural Network (CNN) yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

- 1) Sistem absensi otomatis berhasil dibangun menggunakan metode deep learning CNN yang dapat mendeteksi dan mengenali wajah karyawan secara akurat dalam berbagai kondisi pencahayaan seperti terang dan redup.
- 2) Sistem menggunakan library *face_recognition* dengan metode ekstraksi fitur berbasis *face embedding* dan membandingkan vektor wajah menggunakan jarak Euclidean dengan ambang batas (threshold) sebesar 0.60. Dari pengujian, sistem mampu mengenali wajah dengan nilai jarak rata-rata di bawah threshold, menunjukkan bahwa sistem bekerja dengan tingkat akurasi yang tinggi.
- 3) Uji coba pengenalan wajah pada wajah yang terdaftar seperti "Bahrul Ulum" menunjukkan bahwa semua nilai distance berkisar antara 0.14 hingga 0.29, dan semuanya berhasil dikenali dengan benar, yang membuktikan bahwa sistem dapat mengenali wajah yang sama secara konsisten.
- 4) Pengujian terhadap wajah yang tidak terdaftar menunjukkan hasil jarak di atas threshold (misalnya 0.61 hingga 0.74), yang artinya sistem tidak mengenali wajah asing, sehingga sistem memiliki keamanan yang cukup baik terhadap input tidak sah.
- 5) Sistem telah dilengkapi fitur verifikasi seperti:
 - Notifikasi "Selamat datang [nama]" jika wajah baru pertama kali absen.
 - Notifikasi "Sudah absen" jika wajah sudah pernah absen di hari yang sama.
 - Notifikasi "Wajah tidak dikenali" jika wajah tidak ditemukan dalam database. Hal ini meningkatkan interaksi sistem yang intuitif dan user-friendly.
- 6) Dengan demikian, sistem ini terbukti dapat digunakan sebagai pengganti metode absensi konvensional yang masih rentan terhadap kecurangan dan keterlambatan pencatatan. Sistem otomatis ini mampu mempercepat

proses absensi, meningkatkan akurasi pencatatan, dan meminimalisir keterlibatan manual.

6.2 Saran

Untuk pengembangan lebih lanjut terhadap sistem yang telah dibuat, beberapa saran berikut disampaikan agar sistem menjadi lebih baik dan adaptif di masa mendatang:

- 1) Penambahan jumlah data latih:
Penggunaan lebih banyak variasi wajah dari sudut dan ekspresi yang berbeda akan meningkatkan akurasi pengenalan terutama pada kondisi real-world seperti wajah dengan masker, kacamata, atau pencahayaan ekstrem.
- 2) Optimasi waktu deteksi:
Perlu dilakukan pengujian performa sistem pada perangkat dengan spesifikasi berbeda serta optimasi kecepatan proses deteksi agar sistem dapat digunakan secara real-time dalam skala besar.
- 3) Integrasi dengan perangkat keras lainnya:
Seperti pintu otomatis, sistem keamanan, dan perangkat RFID, untuk mendukung keperluan perusahaan yang lebih kompleks dan menyatu dengan sistem informasi lainnya.
- 4) Penggunaan basis data cloud:
Sistem dapat ditingkatkan dengan menyimpan data absensi ke server berbasis cloud agar dapat diakses dari mana saja dan untuk menghindari kehilangan data lokal.
- 5) Peningkatan keamanan dan privasi:
Implementasi enkripsi pada penyimpanan data wajah dan penerapan sistem login admin yang lebih kuat untuk melindungi data pribadi pengguna.
- 6) Uji coba dalam berbagai lingkungan nyata:
Perlu dilakukan pengujian di berbagai lokasi perusahaan dengan jumlah karyawan yang lebih besar agar sistem terbukti robust dan scalable.

Daftar Pustaka

(JoSDIS), J. o. (2022). Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Labuhanbatu. *Journal of Student Development Information System (JoSDIS)*, Volume 2, Nomor 2, halaman 123.

-
- Ahmad, T. A. (2020). *Jurnal Komputasi*, 8(2), 117-124. *Implementasi Sistem Pengenalan Wajah Menggunakan Deep Learning untuk Verifikasi Identitas Otomatis.*
- Basis Data*. (2017). Informatika Bandung.
- Budiarto Hadiprakoso, R. &. (2022). *Sistem Presensi Menggunakan Face Recognition Berbasis Raspberry Pi*, *jurnal Rekayasa Informasi*, 11(2), 115-124.
- Cahyono, B. (2020). *Belajar Deep Learning dengan Keras*. Andi.
- Devi, S. R. (2022). *Digital Image Processing*. BP Publications.
- Diana, M. ((2019)). *Deep Learning: Konsep dan Aplikasi*. Elex Media Komputindo.
- Erlinda, L. (2019). (Skripsi, Universitas Islam Negeri Sultan Syarif Kasim Riau). *Sistem Presensi Mahasiswa Berbasis Pengenalan Wajah Menggunakan Metode Local Binary Pattern Histogram (LBPH)*.
- Fajri, K. A. (2020). *Implementasi Face Recognition Menggunakan Metode Convolutional Neural Network (CNN) pada Sistem Presensi.*, 7(3), 481-488.
- Firdausi, R. P. (2024). *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, . *Perancangan Sistem Informasi E-Commerce Toko Pakaian Menggunakan Konsep ERD*, 10(1), 123-130.
- Gupta, D. (2013). *Image segmentation Keras: Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras*, 2307.13215.
- Han, H. G. (2020). *Deep Learning for Face Recognition*. Springer.
- Ian Goodfellow, Y. B. (2016). MIT.
- Jusia, P. A. (2020). *JURNAL MEDIA INFORMATIKA BUDIDARMA*. *Sistem Manajemen Absensi dengan Fitur Pengenalan Wajah dan GPS Menggunakan YOLO pada Platform Android*, 4(4), 1235-1242.
- Jyotsana, S. (2023). *Metric Learning for Machine Learning and Computer Vision*. CRC Press.
- Kaya, M. &. (2019). *Symmetry*. *Deep Metric Learning*, 11(9), 1066.

Pengembangan Sistem Deteksi Wajah Menggunakan Deep Learning Untuk Aplikasi Absensi Otomatis Di Perusahaan Aris Motor

- Kendall, K. E. (2011). *Systems Analysis and Design (8th ed.)*. Pearson Education.
- Komputer, W. T. (2017, 2022). *Basis Data MySQL, Panduan Praktis MySQL untuk Pemula*. Andi, Guna Pena.
- Krizhevsky, A. S. (2012). *Imagenet classification with deep convolutional neural networks*. *Advances in Neural Information Processing Systems*, 25.
- Lecun, Y. B. (2015). *Deep learning*. *Nature*, 521(7553), 436–444.
- Li, S. Z. (2019). *Handbook of Face Recognition*. Springer.
- Nurfadillah, A. F. (2020). *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, *Rancang Bangun Sistem Presensi Dosen Menggunakan Face Recognition Berbasis Raspberry Pi*, 6(3), 253-261.
- Overflow., S. (2022).
- Ramadhan, T. (2020). *Skripsi, Universitas Islam Negeri Sultan Syarif Kasim Riau. Sistem Pengenalan Wajah Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Raspberry Pi*.
- Sabili, S. A. (2021). *Jurnal Ilmiah FIFO*, *Implementasi Deep Metric Learning Menggunakan Triplets Loss untuk Pengenalan Wajah*, 13(2), 161-170.
- Sardi, S. J. (2024). *Algoritma: Jurnal Ilmu Komputer dan Informatika*, *Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel*, 8(1), 67-73.
- Satzinger, J. W. (2020). *Systems Analysis and Design in a Changing World*. Cengage Learning.
- Setiono, B. (2020). *Python untuk Pemula*. Elex Media Komputindo.
- Shelly, G. B. (2011). *Systems Analysis and Design (9th ed.)*. Cengage Learning.
- Standisyah, E. &. (2017). *Unisda Journal of Mathematics and Computer*. *IMPLEMENTASI PHPMYADMIN PADA RANCANGAN SISTEM PENGADMINISTRASIAN*, 3(2), 39-44.

- Supriyanto, A. (2016). *Pengenalan Konsep Basis Data*. Andi.
- Thammaiah, R. K. (2021). *ournal of Imaging. Deep Metric Learning for Image Classification: A Survey*, 7(12), 263.
- Utamingrum, F. (2017). *Pengolahan Citra Digital*. Andi.
- Wiatowski, T. S. (2018). *Introduction to Deep Learning*. Springer.
- Wijaya, R. S. (2021). *Wijaya, R., Santoso, H., & Ramadhani, A. Informatika*.
- Yuliana, T. &. (2019). *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA),. Sistem Presensi Berbasis Pengenalan Wajah Menggunakan Metode Viola-Jones dan Eigenface.*, 8(1), 1-8.
- Zhang, K. Z. (2016). *Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks*.
- Zihamussholihin, M. H. (2021). *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA), . Sistem Presensi Pegawai Berbasis Web Menggunakan Metode Face Recognition*, 10(1), 1-9.