BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi di era digital telah memberikan dampak besar pada berbagai aspek kehidupan manusia, termasuk dalam pengelolaan sumber daya manusia. Salah satu penerapan teknologi yang kian diminati adalah sistem absensi berbasis pengenalan wajah. Teknologi ini memberikan beragam manfaat, seperti penghematan waktu, peningkatan ketepatan data, serta meminimalkan peluang terjadinya kecurangan dalam pencatatan kehadiran

Pengenalan wajah merupakan teknologi yang memungkinkan identifikasi individu berdasarkan fitur wajah yang unik. Teknologi ini menggunakan algoritma kompleks untuk menganalisis dan membandingkan gambar wajah seseorang dengan data yang tersimpan dalam basis data. Salah satu metode yang umum digunakan dalam teknologi pengenalan wajah adalah YOLO (You Only Look Once). YOLO merupakan algoritma deteksi objek yang sangat cepat dan efisien, mampu mendeteksi serta mengenali objek pada gambar atau video secara real-time.

Implementasi sistem absensi berbasis *real-time face recognition* menggunakan YOLOv8 diharapkan dapat mengatasi berbagai kelemahan yang ada pada metode konvensional. Dengan sistem ini, proses absensi dapat dilakukan secara otomatis, cepat, dan akurat, sehingga mengurangi potensi kesalahan dan kecurangan. Selain itu, sistem ini juga memberikan kemudahan dalam pengelolaan data kehadiran siswa secara digital, yang dapat diakses dan dipantau secara real-time.

Oleh karena itu, skripsi ini disusun dengan tujuan untuk mengembangkan PENGEMBANGAN SISTEM ABSENSI BERBASIS *REAL-TIME FACE RECOGNITION* DI

LINGKUNGAN SEKOLAH MENGGUNAKAN *YOU ONLY LOOK ONCE* (YOLO) Diharapkan melalui implementasi sistem ini, proses absensi di sekolah dapat menjadi lebih efisien dan akurat.

B. Rumusan Masalah

Berdasarkan uraian latar belakang tersebut, maka perumusan masalah dapat disajikan sebagai berikut:

1. Bagaimana cara mendeteksi wajah siswa dengan menggunakan YOLOv8?
2. Bagaimana membangun sebuah sistem yang dapat mengenali wajah seseorang secara *Real Time*?
3. Bagaimana cara menampilkan data absensi siswa pada *database*?

C. Tujuan Penelitian

Adapun tujuan dari rumusan masalah yang bisa diambil dari penelitian kali ini yaitu :

1. Membuat sistem Mendeteksi Wajah Siswa Menggunakan YOLOv8.
2. Membangun Sistem Pengenalan Wajah Secara *Real-Time*.
3. Menampilkan data absensi siswa pada *database.*

D. Batasan Masalah

Pembatasan masalah pada sebuah penelitian ini adalah :

1. Penelitian ini hanya berfokus pada pengembangan sistem absensi *Real-Time face recognition* menggunakan YOLOv8.
2. Sistem ini akan diuji pada jumlah siswa yang terbatas. Skalabilitas sistem dalam menangani jumlah siswa yang lebih besar perlu diuji lebih lanjut untuk memastikan performa tetap optimal.

E. Manfaat Penelitian

Penelitian tentang pengembangan sistem absensi berbasis *real-time face recognition* di lingkungan sekolah menggunakan *you only look once* (yolo) memiliki berbagai manfaat yang signifikan, antara lain:

1. Manfaat bagi penulis

   Penulis dapat mengetahui cara kerja sistem absensis berbasis *real-time* menggunakan YOLOv8 berdasarkan pengetahuan yang diperoleh selama menjalani pendidikan di Universitas muhammadiyah parepare.

2. Manfaat bagi akademik

   Penelitian ini dapat dijadikan bahan referensi bagi penelitian selanjutnya yang berkaitan dengan *computer vision* dan YOLOv8.

3. Manfaat bagi masyarakat

   Penelitian ini dapat menjadi solusi bagi sekolah dalam memantau murid-murid agar tidak terjadinya kecurangan bagi murid yang sering bolos.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Hasil Penelitian Sebelumnya

Sebagai acuan dalam penelitian ini, disertakan beberapa studi terdahulu yang telah dilakukan oleh peneliti lain :

1. Aplikasi absensi karyawan menggunakan *geolocation* dan *fingert print* berbasis android. (Anggita arfina arfah,2022).

   teknologi absensi berbasis *geolocation* dan *fingerprint* memiliki keunggulan dalam hal akurasi dan keamanan, meskipun menghadapi tantangan dalam hal privasi dan keterbatasan teknis.

2. Sistem pengenalan wajah menggunakan motede YOLO. (Muhammad Syarif Hidayatulloh, 2021).

   penggunaan metode YOLO dalam sistem pengenalan wajah menunjukkan potensi besar dalam aplikasi real-time dengan akurasi tinggi, namun juga memerlukan perhatian khusus terhadap kebutuhan perangkat keras.

3. Impelemtasi *face recognition* pada sistem absensi berbasis android menggunakan metode *convolutional neural network* (CNN) dan *LOCK GPS (*Alfeus adi Saputra, 2022),

   Menggunakan metode CNN dalam sistem pengenalan wajah, namun membutuhkan banyak data latih, proses pelatihan (training) yang memakan waktu, dan overfitting yang memungkinkan Overfitting dapat terjadi dikarenakan terlalu banyak data latih maka algoritma kehilangan kemampuan untuk menggeneralisasi

B. Absensi

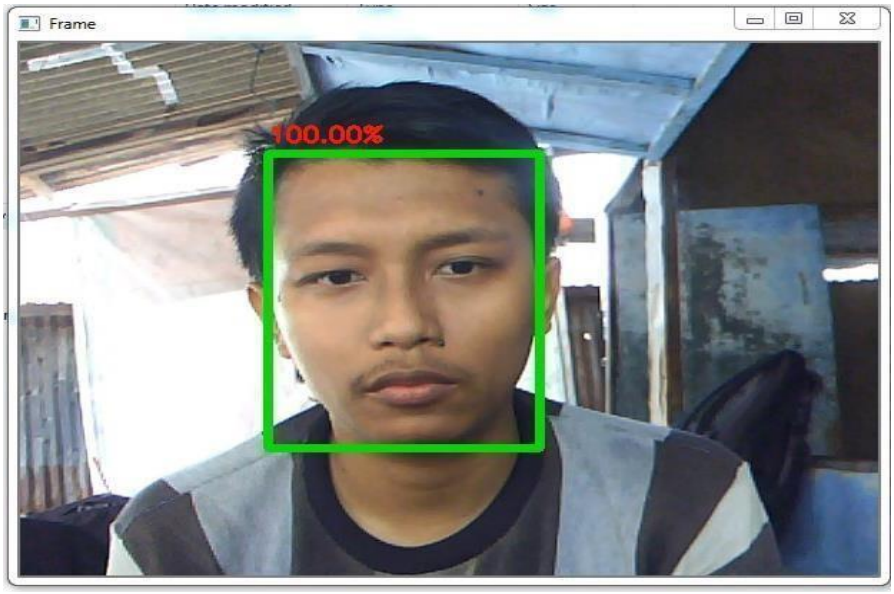
Absensi adalah kegiatan pencatatan dan pelaporan kehadiran siswa yang berperan penting dalam mendukung proses belajar mengajar (Mansur et al., 2020). Kegiatan ini bertujuan untuk memperoleh informasi terkait kehadiran siswa, khususnya dalam kegiatan pembelajaran, sekaligus menjadi bukti bahwa siswa telah mengikuti pelajaran. Absensi berkaitan erat dengan aspek penilaian dan kedisiplinan siswa, karena umumnya dilakukan sebelum pelajaran dimulai sehingga siswa yang datang terlambat dianggap tidak hadir. Selain itu, absensi juga dapat dijadikan indikator apakah proses pembelajaran berlangsung sebagaimana mestinya. Bahkan, menurut Pulungan & Saleh (2019), absensi dapat dianggap sebagai salah satu tolok ukur utama dalam menilai siswa, sebab permasalahan kehadiran akan berdampak langsung pada hasil belajar mereka."

Teknologi mesin absensi wajah merupakan sistem yang mampu mengidentifikasi atau memverifikasi identitas seseorang melalui wajah, yang umum diterapkan pada metode absensi berbasis pemindaian wajah. Teknologi ini dapat mengenali wajah dari foto, video, maupun secara langsung. Menggunakan teknologi biometrik, mesin absensi wajah mampu mendeteksi karakteristik biologis manusia, salah satunya melalui fitur face recognition atau pendeteksian wajah. Dengan penerapan teknologi ini, absensi karyawan dapat dilakukan secara unik, akurat, dan sulit untuk dipalsukan.

C. Deteksi Wajah

Ketika program dijalankan, sistem akan mengaktifkan kamera yang menangkap objek beserta lingkungan sekitarnya, sehingga beban komputasi yang diperlukan menjadi cukup besar. Untuk mengurangi beban tersebut, diperlukan penentuan batas perhitungan yang berfungsi membatasi area pemrosesan. Proses deteksi wajah ini juga bertujuan untuk melokalisasi area wajah secara fleksibel, sehingga mampu mengidentifikasi setiap input di seluruh area frame dan menentukan apakah input tersebut merupakan wajah atau bukan (Sukusvieri, 2020). Proses deteksi wajah tersebut ditunjukkan pada Gambar 2.1 :

Gambar 2.1 Deteksi wajah (Sukusvieri, 2020)



Sistem pengenalan wajah yang lazim digunakan saat ini dapat dikelompokkan sebagai berikut:

1. Berbasis penampilan, algoritma bekerja dengan memperhatikan keseluruhan citra wajah tanpa menitikberatkan pada struktur geometris maupun sifat reflektansinya.
2. Berbasis fitur, algoritma mengandalkan bagian-bagian tertentu dari wajah, seperti hidung, mata, dan mulut, sebagai elemen utama untuk proses pengenalan.
3. Hibrida, di mana algoritma kompleks berdasarkan campuran dari dua yang disebutkan di atas, yang dibuat untuk mengurangi adanya tingkat kecocokan palsu.

Beberapa algoritma populer yang digunakan dalam klasifikasi wajah antara lain Eigenfaces dengan metode Principal Component Analysis (PCA), yang berfungsi mereduksi dimensi citra wajah sehingga menghasilkan jumlah variabel yang lebih sedikit dan lebih mudah dianalisis. Selain itu, terdapat algoritma Support Vector Machine (SVM) yang menerapkan teknik klasifikasi pada data linear maupun non-linear. Algoritma lainnya adalah K-Nearest Neighbour (KNN), yaitu metode klasifikasi objek berdasarkan data latih dengan jarak terdekat terhadap objek yang dianalisis. Sementara itu, Convolutional Neural Network (CNN) merupakan jenis jaringan saraf tiruan yang paling sering digunakan untuk memproses dan menganalisis citra visual.

Sistem pengenalan wajah umumnya memiliki dua tahap utama, yaitu identifikasi dan verifikasi, dengan urutan proses algoritma sebagai berikut:

1. Deteksi wajah, algoritma menandai dan mengidentifikasi area wajah pada citra yang diproses.
2. Deteksi fitur wajah, algoritma menghitung titik-titik antropometri pada wajah sebagai acuan pengenalan.
3. Normalisasi wajah, sistem melakukan transformasi tambahan, seperti menghilangkan kemiringan kepala dan menyesuaikan warna kulit, guna memperoleh citra wajah frontal yang jelas.
4. Ekstraksi fitur dan perhitungan deskriptor, sistem menghasilkan deskriptor, yakni sekumpulan karakteristik unik wajah yang independen dari faktor usia, gaya rambut, maupun riasan. Perbandingan antar-deskriptor digunakan untuk menentukan apakah dua citra wajah berasal dari orang yang sama.
5. Verifikasi, template digital yang telah dibuat dibandingkan dengan data wajah yang tersimpan di basis data untuk menyelesaikan proses identifikasi.

D. YOLOv8

Penelitian ini menerapkan deteksi objek untuk mengenali wajah siswa dengan menggunakan metode YOLO (You Only Look Once). YOLO merupakan salah satu metode mutakhir (state-of-the-art) dalam pendeteksian objek secara real-time. Metode ini bekerja sebagai detektor dengan model terpadu (unified) yang secara langsung memprediksi kotak pembatas dan probabilitas kelas pada keseluruhan citra hanya dalam satu kali evaluasi (H et al., 2020).

You Only Look Once (YOLO) adalah sistem deteksi berbasis Convolutional Neural Network (CNN). Dalam metode ini, satu jaringan convolutional secara bersamaan memprediksi sejumlah bounding box beserta probabilitas kelas objek di dalamnya. YOLO dilatih untuk menganalisis keseluruhan gambar, sehingga mampu meningkatkan kinerja deteksi objek. Dengan menggunakan satu jaringan convolutional dalam proses pendeteksian, YOLO memiliki beberapa keunggulan dibandingkan metode deteksi objek lainnya.

Berikut adalah beberapa keunggulan motede YOLOv8 :

1. Performa tinggi

   YOLOv8 menawarkan kinerja yang lebih unggul dibandingkan versi sebelumnya, dengan kemampuan deteksi objek yang lebih cepat dan akurat.

2. Model yang lebih kecil

   Ukuran model YOLOv8 lebih kecil dibandingkan generasi sebelumnya, sehingga lebih praktis dan mudah digunakan

3. Ekstraksi fitur yang lebih baik

   Menggunakan metode ekstraksi fitur yang lebih canggih, YOLOv8 mampu menghasilkan deteksi objek dengan akurasi yang lebih baik.

4. Kemampuan multi-skala

   YOLOv8 dapat mendeteksi objek dengan berbagai ukuran dalam satu gambar, meningkatkan fleksibilitas deteksi.

5. Kemampuan deteksi banyak objek

   Mampu mendeteksi banyak objek sekaligus dalam satu gambar, sehingga mempercepat proses analisis.

6. Kemampuan deteksi objek

   YOLOv8 dapat menangani gambar berukuran besar dengan akurasi tinggi, memudahkan proses deteksi objek

E. Python

Python adalah bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada akhir 1980-an (Kurnia & Rahmi, 2022:3). Bahasa ini sering digunakan dalam pembuatan program berbasis OpenCV karena memiliki sintaks yang lengkap namun tetap sederhana dan mudah dipahami.

Selain mudah dipelajari, Python juga memiliki berbagai keunggulan lain yang membuatnya banyak digunakan oleh programmer dan pengembang web. Adapun beberapa keunggulan Python yang dirasakan oleh penggunanya adalah sebagai berikut:

1. Mendukung berbagai platform – Python dapat dijalankan pada beragam sistem operasi komputer, seperti Windows, macOS, Linux, Raspberry Pi, dan lainnya.
2. Sintaks berbasis bahasa Inggris – Struktur sintaks Python menggunakan bahasa Inggris, sehingga mudah diingat, dibaca, dan dipelajari, bahkan bagi pengguna tanpa latar belakang pemrograman.
3. Sintaks ringkas – Dibandingkan bahasa pemrograman lain, Python memungkinkan penulisan kode yang lebih sederhana dan tidak terlalu panjang.

4. Interpreter bawaan – Python memiliki sistem interpreter yang dapat mengeksekusi kode secara langsung setelah ditulis, sehingga proses pembuatan prototipe menjadi lebih cepat.
5. Mendukung berbagai paradigma pemrograman – Python dapat digunakan untuk pemrograman prosedural, berorientasi objek, maupun fungsional.
6. Penulisan kode lebih sederhana – Berbeda dengan beberapa bahasa lain yang memerlukan tanda semicolon (;) atau tanda kurung untuk mengakhiri perintah, Python cukup menggunakan baris baru.
7. Penentuan cakupan yang sederhana – Jika pada bahasa lain biasanya digunakan tanda kurung kurawal untuk menentukan scope, di Python cukup menggunakan indentasi (spasi dan paragraf).



```python
# let's start with the Imports
import cv2
import numpy as np

# Read the image using imread function
image = cv2.imread('image.jpg')
cv2.imshow('Original Image', image)

# let's downscale the image using new  width and height
down_width = 300
down_height = 200
down_points = (down_width, down_height)
resized_down = cv2.resize(image, down_points, interpolation= cv2.INTER_LINEAR)

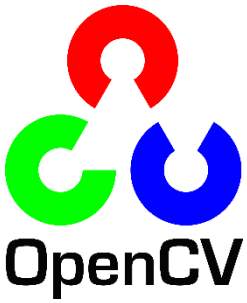
# let's upscale the image using new  width and height
up_width = 600
up_height = 400
up_points = (up_width, up_height)
resized_up = cv2.resize(image, up_points, interpolation= cv2.INTER_LINEAR)

# Display images
cv2.imshow('Resized Down by defining height and width', resized_down)
cv2.waitKey()
cv2.imshow('Resized Up image by defining height and width', resized_up)
cv2.waitKey()

#press any key to close the windows
cv2.destroyAllWindows()
```

Gambar 2.2 Source Code using OpenCV Python.png
(Sumber : Source_ dezyre.gumlet.io)

F. OpenCV



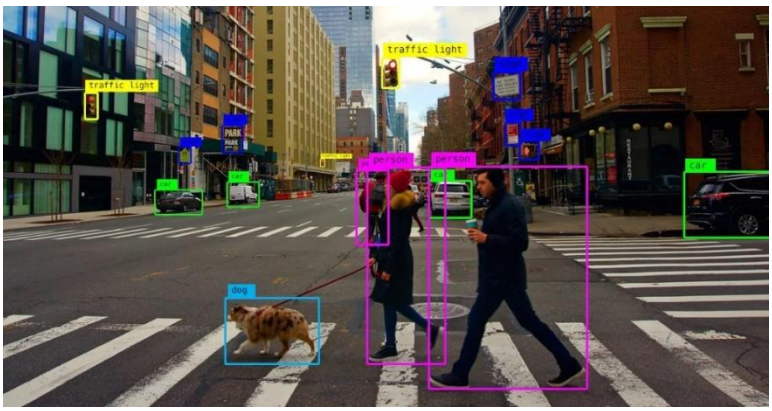
Gambar 2.3 OpenCV
(Sumber : iconduck.com)

OpenCV, atau *Open Source Computer Vision*, merupakan pustaka berbasis *open source* yang menyediakan lebih dari 500 fungsi untuk keperluan *computer vision*. Perangkat lunak ini dirilis di bawah lisensi BSD, sehingga dapat digunakan baik untuk keperluan bisnis maupun komersial. OpenCV awalnya dikembangkan menggunakan bahasa pemrograman C, namun juga menyediakan antarmuka untuk bahasa pemrograman lain seperti Python guna memudahkan akses dan pemanfaatannya (Kadir, 2019).

OpenCV menyediakan beragam fitur dan fungsionalitas untuk pengolahan citra serta analisis video, di antaranya:

1. Pembacaan dan penulisan citra/video – Mendukung pembacaan serta penyimpanan citra dan video dalam berbagai format file, seperti BMP, JPEG, PNG, dan lainnya.
2. Praproses citra – Menyediakan teknik praproses seperti penyesuaian kontras dan kecerahan, filtrasi, kompresi, pengaburan (blurring), dan berbagai metode lainnya.
3. Deteksi objek – Memiliki dukungan untuk deteksi objek dengan berbagai algoritma, termasuk deteksi wajah, deteksi gerakan, dan deteksi tepi.
4. Ekstraksi fitur – Menawarkan algoritma untuk mengekstrak fitur citra, seperti deskriptor histogram warna, deskriptor tekstur, dan lainnya.
5. Pelacakan objek – Menyediakan metode pelacakan objek dalam video bergerak, baik berbasis gerakan, warna, maupun pendekatan lainnya.

6. Kalibrasi kamera – Memungkinkan kalibrasi kamera guna memperkirakan parameter intrinsik dan ekstrinsik, yang penting untuk aplikasi computer vision berbasis citra 3D.
7. Visi stereo – Mendukung pengolahan citra stereo untuk estimasi kedalaman dan rekonstruksi 3D.



G. Computer Vision

Gambar 2.4 Computer vision
(Sumber : metanesia.id)

Computer Vision adalah kemampuan suatu mesin atau komputer untuk melihat dan mengenali citra, setara bahkan melampaui kemampuan penglihatan manusia. Pengolahan citra merupakan salah satu teknologi yang digunakan untuk memecahkan permasalahan terkait pemrosesan gambar, di mana gambar diolah sedemikian rupa agar lebih mudah diproses. Sementara itu, tujuan utama computer vision adalah menghasilkan keputusan yang bermanfaat mengenai objek fisik nyata berdasarkan data yang diperoleh dari perangkat atau sensor (T. Susim, 2021).

Computer Vision adalah teknologi yang memungkinkan komputer melihat dan mengenali objek di sekitarnya layaknya manusia, dengan bantuan kamera dan kecerdasan buatan (*Artificial Intelligence*). Berbeda dengan manusia, komputer tidak mengalami kelelahan, sehingga mampu menganalisis objek dalam jumlah besar dalam waktu singkat. Kemampuan ini memungkinkan komputer mendeteksi cacat pada produk yang mungkin luput dari penglihatan manusia.

Computer Vision memerlukan basis data yang besar agar dapat berfungsi secara optimal, karena dibutuhkan banyak informasi untuk mengantisipasi setiap kemungkinan sesuai dengan tugas yang diberikan. Misalnya, sebuah komputer yang dilatih untuk mengenali tanaman sehat harus 'melihat' ribuan data referensi visual, mencakup gambar tanaman, lahan pertanian, hewan, dan objek terkait lainnya. Dengan kumpulan data yang memadai, komputer dapat mengenali berbagai jenis tanaman sehat secara akurat, membedakannya dari tanaman yang tidak sehat, menilai kualitas lahan, mendeteksi hama, maupun mengidentifikasi keberadaan hewan di sekitar tanaman.

Berikut ini adalah contoh mengaplikasikan Computer Vision yang menunjukkan potensi AI untuk membantu industri :

1. Manufaktur

   Dalam industri manufaktur, Computer Vision digunakan untuk otomatisasi inspeksi kualitas produk di lini produksi, membantu mendeteksi cacat secara real-time sehingga meningkatkan efisiensi dan mengurangi limbah.

2. Ritel

   Di sektor ritel, teknologi Computer Vision membantu menganalisis perilaku pelanggan melalui kamera CCTV, mengoptimalkan penempatan produk, serta memungkinkan kasir otomatis dengan sistem checkout tanpa kasir.

3. Kesehatan

   Dalam bidang kesehatan, Computer Vision mendukung diagnosis medis melalui analisis citra radiologi seperti X-ray, MRI, dan CT scan, membantu dokter mendeteksi penyakit lebih awal dan lebih akurat.

4. Transportasi

Di industri transportasi, Computer Vision digunakan untuk deteksi pelanggaran lalu lintas, pemantauan kondisi jalan, serta sistem bantuan mengemudi cerdas (ADAS) yang mendukung kendaraan otonom.

5. Pertanian

   Dalam pertanian, Computer Vision diaplikasikan untuk mendeteksi kondisi tanaman, mengidentifikasi hama dan penyakit, serta membantu proses panen otomatis menggunakan drone dan robot.

6. Keamanan dan Pertahanan

   Di sektor keamanan, Computer Vision digunakan dalam sistem pengenalan wajah, pelacakan objek mencurigakan, dan pengawasan area sensitif secara otomatis, meningkatkan respons terhadap ancaman.

H. Visual Studio Code



Gambar 2.5 Visual Studio Code Logo
(Sumber : logowik.com)

Visual Studio Code adalah sebuah *text editor* yang ringan namun andal, dikembangkan oleh Microsoft untuk sistem operasi multiplatform, sehingga tersedia untuk Linux, MacOS, dan Windows. Editor ini secara bawaan mendukung bahasa pemrograman JavaScript, TypeScript, dan Node.js. Selain itu, bahasa pemrograman lainnya seperti C++, C#, Python, Go, Java, dan PHP dapat digunakan melalui pemasangan *plugin* yang tersedia di *marketplace* Visual Studio Code (Ummy Gusti Salamah, 2021).

Visual Studio Code secara bawaan mendukung JavaScript, TypeScript, dan Node.js. Selain itu, editor ini memiliki ekosistem *extension* yang luas untuk mendukung berbagai bahasa pemrograman dan *runtime* lainnya, seperti C++, C#, Java, Python, PHP, Go, dan .NET. Dengan fleksibilitas serta ketersediaan *extension* yang beragam, Visual Studio Code menjadi salah satu pilihan favorit di kalangan pengembang untuk berbagai jenis tugas pemrograman.

Fitur Penting VS Code diantaranya adalah :

1. IntelliSense merupakan fitur pintar untuk *code completion* yang secara otomatis memberikan saran berdasarkan konteks saat pengetikan, sehingga membantu pengembang menulis kode lebih cepat, efisien, dan dengan tingkat kesalahan yang lebih rendah.
2. Run and Debug adalah fitur bawaan Visual Studio Code yang memungkinkan pengembang menjalankan serta melakukan *debugging* kode untuk berbagai bahasa pemrograman. Fitur ini mendukung penentuan *breakpoint*, pemeriksaan variabel, dan penelusuran kode, sehingga memudahkan identifikasi serta perbaikan masalah.
3. Built-in Git adalah fitur integrasi Git di Visual Studio Code yang memungkinkan pengembang mengelola version control langsung dari dalam editor. Dengan fitur ini, pengembang dapat mengatur, memantau, dan membuat perubahan pada repositori tanpa perlu beralih ke aplikasi Git terpisah.
4. Extensions adalah pustaka ekstensi yang tersedia di marketplace Visual Studio Code. Dengan ekstensi ini, pengembang dapat menambahkan dukungan untuk berbagai bahasa pemrograman, *framework*, dan alat lain, sehingga editor dapat disesuaikan dengan kebutuhan spesifik proyek.
5. Tema yang dapat disesuaikan: Visual Studio Code menyediakan berbagai pilihan tema untuk mengubah tampilan dan skema warna editor, sehingga dapat disesuaikan dengan

preferensi pengguna serta membantu mengurangi ketegangan mata selama sesi pemrograman yang panjang.

I. Blackbox Testing

Black-box testing bertujuan untuk menguji seluruh fungsi dari suatu program aplikasi. Pengujian ini dilakukan untuk memastikan bahwa program berfungsi sesuai dengan yang diinginkan, tanpa memerlukan pengetahuan tentang kode sumber yang digunakan. Metode ini berfokus pada pengujian input dan output sistem guna memverifikasi kesesuaian fungsi program.

Black-box testing, atau dikenal juga sebagai behavioral testing, merupakan metode pengujian yang memeriksa hasil input dan output perangkat lunak tanpa mengetahui struktur kodenya. Pengujian ini biasanya dilakukan pada tahap akhir pengembangan untuk memastikan perangkat lunak berfungsi dengan baik.

Terdapat beberapa teknik yang umum digunakan dalam proses pengujian perangkat lunak. Berikut ini adalah beberapa di antaranya:

1. All pair testing

   Teknik all pair testing ini dikenal juga dengan pairwise testing. Pengujian ini digunakan untuk menguji semua kemungkinan kombinasi dari seluruh pasangan berdasarkan input parameternya.

2. Boundary value analysis

   Teknik ini berfokus pada pencarian kesalahan baik dari sisi eksternal maupun internal perangkat lunak, sehingga dapat memastikan bahwa setiap kombinasi input yang diuji memberikan output yang sesuai dengan yang diharapkan.

3. Equivalence partitioning

   Teknik ini bekerja dengan cara membagi data input dari perangkat lunak menjadi beberapa partisi atau kelompok, di mana setiap partisi mewakili himpunan data yang

diharapkan memberikan perilaku atau hasil yang sama. Dengan begitu, pengujian dapat dilakukan lebih efisien tanpa harus menguji semua kemungkinan input.

4. Cause-effect graph

   Teknik cause-effect graph adalah metode pengujian yang menggunakan grafik untuk memvisualisasikan hubungan antara penyebab (cause) dan akibat (effect) dari suatu kesalahan atau perilaku perangkat lunak. Grafik ini membantu tester mengidentifikasi kondisi input yang dapat memicu keluaran tertentu, sehingga memudahkan dalam menemukan potensi error yang mungkin terjadi.

5. Fuzzing

   Fuzzing adalah teknik pengujian perangkat lunak yang bertujuan menemukan bug atau celah keamanan dengan cara memasukkan data yang tidak valid, acak, atau tidak sempurna ke dalam sistem. Teknik ini digunakan untuk menguji ketahanan perangkat lunak terhadap input yang tidak terduga, sehingga dapat membantu mengidentifikasi potensi kerentanan atau error sebelum perangkat lunak digunakan secara luas.

6. Orthogonal array testing

   Orthogonal Array Testing adalah teknik pengujian perangkat lunak yang digunakan untuk menguji kombinasi input dengan jumlah relatif kecil, tetapi tetap memberikan cakupan pengujian yang luas. Teknik ini menggunakan konsep *orthogonal array* (tabel atau matriks khusus) untuk memastikan setiap pasangan nilai input diuji secara merata. Metode ini efisien untuk menemukan defect dengan jumlah kasus uji yang minimal, terutama jika digunakan pada sistem dengan variabel input yang sedikit namun kompleks.

7. State transition

State Transition Testing adalah teknik pengujian yang digunakan untuk memverifikasi perilaku sistem berdasarkan perubahan status (*state*)., teknik ini membantu memastikan bahwa setiap transisi berjalan sesuai alur yang diharapkan.

J. Whitebox Testing

Blackbox testing bertujuan untuk menguji seluruh fungsi dari fasilitas program aplikasi. Pengujian dengan menggunakan *black-box* testing di mana pengujian ini hanya bertujuan untuk melihat program tersebut apakah sesuai dengan fungsi yang diinginkan program tersebut tanpa mengetahui kode program yang dipakai.(Efendi, 2021).

White-box testing atau yang sering disebut sebagai *pengujian kotak putih* adalah metode pengujian perangkat lunak yang dilakukan dengan menganalisis serta meneliti struktur internal dan kode program secara langsung. Berbeda dengan *black-box testing* yang hanya berfokus pada pengamatan hasil masukan (*input*) dan keluaran (*output*) tanpa mengetahui kode sumber, *white-box testing* menitikberatkan pada pemeriksaan aliran data, logika, dan struktur program untuk memastikan setiap bagian berjalan sesuai dengan yang diharapkan.

Beberapa teknik yang dapat diterapkan dalam melakukan *white-box testing* pada perangkat lunak antara lain:"

1. Basis path testing

   Basis path testing merupakan teknik pertama yang digunakan untuk mengukur tingkat kompleksitas kode program serta menentukan jalur eksekusi yang akan dijalankan

2. Branch coverage

   Selanjutnya terdapat teknik branch coverage, yaitu metode pengujian yang memastikan setiap cabang kode dieksekusi minimal satu kali.

3. Condition coverage

   Berikutnya ada teknik condition coverage yang bertujuan menguji seluruh kode sehingga setiap kondisi dapat menghasilkan nilai TRUE maupun FALSE. Dengan cara

ini, penguji dapat memastikan perangkat lunak berfungsi dengan baik dan menghasilkan output yang sesuai dengan input pengguna.

4. Loop lesting

   Terdapat pula teknik loop testing, yaitu pengujian yang dilakukan untuk memeriksa berbagai struktur perulangan dalam program seperti do-while, for, dan while. Melalui pengujian ini, kondisi pada setiap perulangan dapat diperiksa untuk memastikan apakah sudah berjalan sesuai dengan yang diharapkan.

5. Multiple condition coverage

   Selanjutnya, terdapat teknik multiple condition coverage, yang digunakan untuk menguji semua kemungkinan kombinasi kode pada berbagai kondisi. Setiap kombinasi harus diuji setidaknya sekali untuk memastikan perangkat lunak berfungsi dengan baik.

6. Statement coverage

   Teknik terakhir adalah statement coverage, yang dilakukan dengan menguji setiap pernyataan dalam perangkat lunak minimal satu kali. Melalui pengujian ini, kesalahan pada kode dapat terdeteksi sehingga dapat segera diperbaiki.

K.Webcam

Webcam merupakan perangkat kamera digital yang dihubungkan ke laptop atau komputer melalui USB, port COM, atau menggunakan jaringan internet dan Wi-Fi. Perangkat ini dirancang untuk menampilkan gambar secara langsung tanpa perlu menyimpannya, karena fungsi utamanya adalah merekam dan mengirimkan gambar secara real-time. Beberapa jenis webcam menggunakan kabel USB yang terpasang di bagian belakang sebagai media penghubung" (Sinta, 2020).

Pada awal kemunculannya, webcam tergolong sebagai perangkat dengan harga yang cukup mahal. Namun, seiring perkembangan teknologi, banyak laptop yang kini sudah dilengkapi dengan webcam pada layarnya. Berbeda halnya dengan komputer desktop yang umumnya tidak memiliki webcam bawaan, sehingga penggunanya perlu membeli perangkat tersebut secara terpisah.



Gambar 2.6 Webcam
(Sumber: Google.com)

L. Unified Modeling Language (UML)

Menurut Riri Fitri Sari dan Ardiati (2021:107), Unified Modeling Language (UML) merupakan bahasa pemodelan yang berfungsi untuk menentukan spesifikasi, memvisualisasikan, membangun, serta mendokumentasikan artefak pada sistem perangkat lunak, baik yang masih dalam tahap perancangan maupun yang sedang dikembangkan.

UML (Unified Modelling Language) merupakan metode pemodelan visual yang digunakan untuk merancang sistem berbasis objek. Bahasa ini pertama kali diperkenalkan oleh Object Management Group dengan versi awal 1.0 pada Januari 1997. UML dapat diartikan sebagai bahasa standar untuk memvisualisasikan, merancang, dan mendokumentasikan sistem, atau sering disebut sebagai bahasa standar dalam penulisan blueprint perangkat lunak. Keberadaan UML diharapkan dapat mempermudah proses pengembangan perangkat lunak

(RPL) serta mampu memenuhi kebutuhan pengguna secara efektif, lengkap, dan tepat, termasuk dalam aspek skalabilitas, ketahanan, keamanan, dan lainnya.

Perlu dipahami bahwa sebuah sistem yang berkualitas berawal dari proses perancangan dan pemodelan yang terencana dengan baik. Salah satu metode yang dapat digunakan adalah UML. Tujuan dan fungsi penggunaan UML antara lain sebagai berikut:

1. Menyediakan bahasa pemodelan visual atau berbentuk gambar yang dapat digunakan oleh pengguna dari berbagai jenis bahasa pemrograman maupun proses umum rekayasa.
2. Mengintegrasikan berbagai informasi terbaik yang tersedia dalam pemodelan.
3. Menyajikan gambaran model atau bahasa pemodelan visual yang ekspresif untuk mendukung pengembangan system.
4. Tidak hanya memvisualisasikan model perangkat lunak, tetapi juga mampu memodelkan sistem berorientasi objek.
5. Memudahkan pengguna dalam memahami dan membaca suatu sistem.
6. Berfungsi sebagai blueprint yang menyajikan informasi lebih detail untuk perancangan, termasuk dalam bentuk kode program.

M. Flowchart

Menurut Sofwan Hanief (2020:8), flowchart adalah metode yang digunakan untuk memvisualisasikan urutan logika dalam suatu prosedur penyelesaian masalah. Dengan kata lain, flowchart menggambarkan tahapan-tahapan penyelesaian masalah yang direpresentasikan melalui simbol-simbol tertentu.

Flowchart atau bagan alur merupakan diagram yang menggambarkan rangkaian langkah dan keputusan dalam menjalankan suatu proses pada program. Setiap langkah direpresentasikan dengan simbol diagram yang saling terhubung melalui garis atau panah. Flowchart memiliki peran penting dalam menentukan langkah atau fungsi dalam proyek

pembuatan program, khususnya yang melibatkan banyak pihak. Dengan bagan alur, proses program menjadi lebih jelas, singkat, dan meminimalkan risiko kesalahpahaman. Dalam dunia pemrograman, flowchart juga berfungsi sebagai jembatan yang menghubungkan kebutuhan teknis dan non-teknis.

Fungsi utama flowchart adalah menggambarkan alur jalannya suatu program dari satu proses ke proses berikutnya, sehingga alur program dapat dipahami dengan mudah oleh siapa saja. Selain itu, flowchart juga berfungsi untuk menyederhanakan rangkaian prosedur, sehingga informasi yang disampaikan menjadi lebih mudah dipahami.

Flowchart memiliki lima jenis utama, di mana setiap jenisnya memiliki ciri khas dan tujuan penggunaan yang berbeda.:

1. Flowchart focument

   Flowchart dokumen, yang juga dikenal sebagai paperwork flowchart, digunakan untuk menggambarkan alur perpindahan formulir dari satu bagian ke bagian lainnya. Diagram ini mencakup proses bagaimana sebuah laporan diproses, dicatat, hingga disimpan.

2. Flowchart program

   Flowchart ini menjelaskan secara detail prosedur dalam suatu proses program. Terdapat dua jenis flowchart program, yaitu flowchart logika program (program logic flowchart) dan flowchart program komputer terperinci (detailed computer program flowchart).

3. Flowchart proses

   Flowchart proses merupakan metode penggambaran dalam rekayasa industri yang digunakan untuk merinci serta menganalisis langkah-langkah berikutnya dalam suatu prosedur atau sistem.

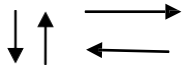
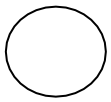
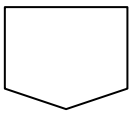
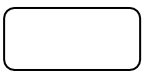
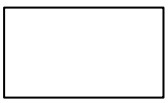
4. Flowchart sistem

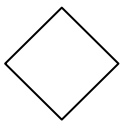
Flowchart sistem adalah diagram yang menggambarkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara keseluruhan. Selain itu, flowchart ini juga menjelaskan urutan setiap prosedur yang terdapat dalam sistem tersebut.

5. Flowchart skematik

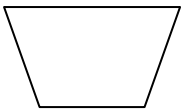
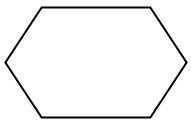
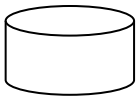
Flowchart ini menggambarkan alur prosedur suatu sistem yang hampir mirip dengan flowchart sistem. Perbedaannya terletak pada penggunaan simbol-simbol untuk menunjukkan alur proses. Selain simbol, flowchart skematik juga memanfaatkan gambar komputer dan peralatan lainnya guna mempermudah pembacaan bagi orang awam. Secara umum, setiap simbol dalam flowchart memiliki makna yang berbeda-beda. Berikut adalah beberapa simbol yang umum digunakan dalam pembuatan flowchart..

Tabel 2.1 Simbol Flowchart

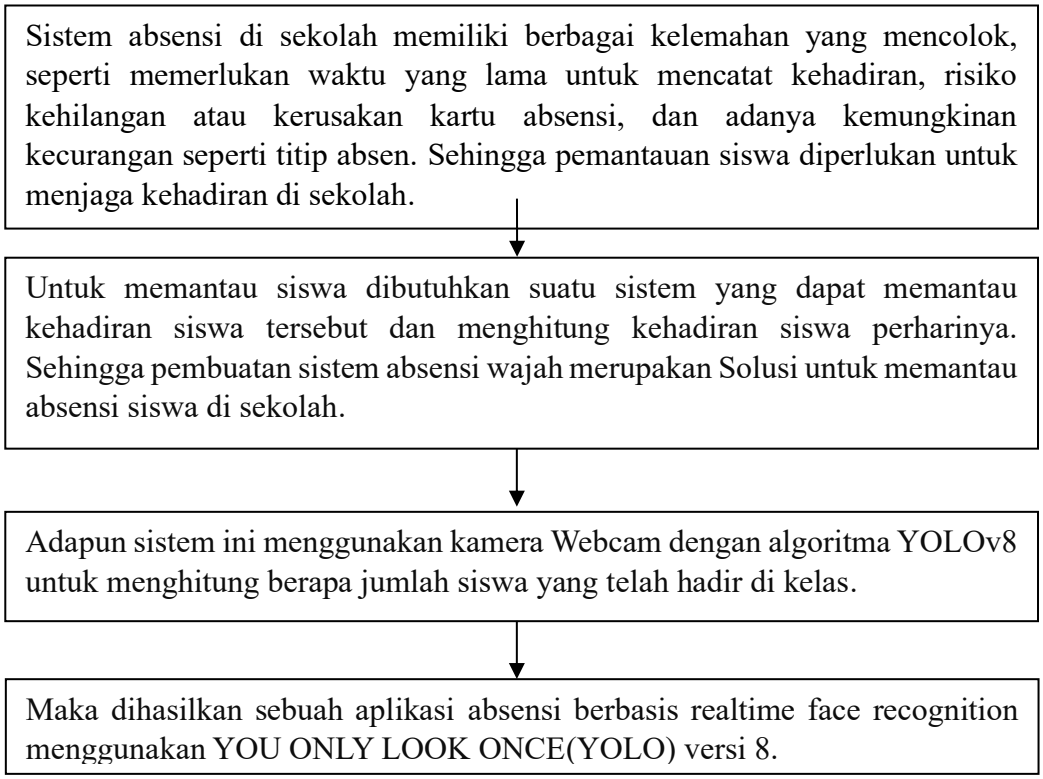
No.	Nama	Simbol	Keterangan
1.	Arus Dokumen/ Pemrosesan		Arah arus dokumen atau pemrosesan; arus normal adalah kekanan atau kebawah.
2.	Penghubung Dalam Sebuah Halaman		Menghubungkan bagian alir pada halaman yang sama. Simbol ini digunakan untuk menghindari terlalu banyak anak panah yang saling melintang.
3.	Penghubung Pada Halaman Berbeda		Menghubungkan bagian alir pada halaman yang berbeda. Simbol ini digunakan untuk menghindari terlalu banyak anak panah yang saling melintang.
4.	Terminal		Digunakan untuk memulai, mengakhiri atau titik henti dalam sebuah proses atau program.
5.	Pemrosesan Komputer		Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer, biasanya menghasilkan data atau informasi.

6.	Keputusan		Sebuah tahap pembuatan keputusan; digunakan dalam bagan alir program komputer untuk menunjukan cabang bagi alternatif cara.

Lanjutan tabel 2.1

No.	Nama	Simbol	Keterangan
7.	Catatan		Digunakan untuk menggambarkan catatan akuntansi yang digunakan untuk mencatat data yang direkam sebelumnya dalam dokumen atau formulir.
8.	Operasi Manual		Menunjukan proses yang dikerjakan secara manual
9.	Dokumen		Sebuah dokumen atau laporan. dokumen dapat dibuat dengan tangan atau cetak oleh komputer.
10.	Preperation		Simbol untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal
11.	Database		Simbol yang menunjukkan penyimpanan data dalam sistem atau proses dan digunakan untuk merepresentasikan tempat atau lokasi penyimpanan data dalam suatu sistem atau database.

N. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penilitian

Penelitian Penelitian ini menggunakan metode eksperimen untuk menguji dan mengembangkan algoritma serta teknik dalam memproses, menganalisis, dan memahami gambar dan video. Langkah-langkah yang terlibat mencakup perancangan eksperimen untuk menguji keandalan algoritma pengenalan objek, pengumpulan dataset, serta analisis hasil untuk mengevaluasi kinerja sistem.

B. Tempat dan Waktu Penelitian

Proses penelitian dilakukan di sekolah UPTD SMKS MUHAMMADIYAH KOTA PAREPARE yang beralamat di jl. Muhammadiyah No.08 Kota Parepare, Waktu yang dibutuhkan dalam pelaksanaan penelitian ini adalah ±2 Bulan.

C. Alat dan Bahan

Adapun alat dan bahan yang akan dibutuhkan dalam penelitian ini sebagai berikut :

1. Alat

   Alat yang digunakan dalam membuat aplikasi ini yaitu laptop dengan spesifikasi sebagai berikut :

   a. RAM 8GB

   b. Processor IntelCore i3

   c. Penyimpanan harddisk 1TB

   d. VGA Nvidia Gforce 920MX

   e. WEBCAM 1080mp

2. Bahan

   Perangkat Lunak yang digunakan dalam membuat aplikasi ini yaitu sebagai berikut :

a. Python
b. Visual studio code
c. OpenCV
d. YOLO
e. Deepface
f. Tensorflow
g. Dataset(foto siswa)

D. Metode Pengumpulan Data

Metode pengumpulan data yang akan digunakan adalah sebagai berikut:

1. Observasi

   Metode pengumpulan data melalui pengamatan langsung. atau peninjauan secara cermat pada sekolah.

2. Wawancara

   Melakukan wawancara dengan narasumber pada suatu sekolah yakni guru atau siswa guna untuk mendapatkan beberapa informasi untuk dijadikan acuan dalam perancangan sistem absensi.

3. Studi literatur

   Pencarian informasi dalam bentuk membaca atau mengambil informasi dari berbagai literatur yang berkaitan dengan judul penelitian contohnya jurnal, artikel, laporan penelitian. setelah data penelitian terkumpul, kemudian dikelolah sehingga dapat diperoleh kesimpulan yang objektif dari peneliti.

E. Tahapan Penelitian

Tahapan penelitian ini akan dilakukan dalam beberapa tahap yaitu:

1. Persiapan penelitian

Persiapan penelitian sebagai berikut, Menyiapkan buku-buku, artikel tentang topik penelitian serta hardware dan software yang digunakan dalam penelitian.

2. Pengumpulan data

   Pengumpulan data dilakukan untuk mengumpulkan informasi melalui studi literatur berupa dataset dan library yang akan digunakan.

3. Perancangan sistem

   Peneliti merancang aplikasi yang berdasarkan pemecahan masalah.

4. Pembuatan sistem

   Sistem akan dibangun berdasarkan data yang telah didapatkan setelah data dikumpulkan.

5. Pengujian

   Setelah melakukan pembuatan sistem maka dilakukan pengujian pada sistem tersebut. Jika hasil terdapat kekurangan atau mengalami kerusakan maka akan diperbaiki.

6. Implementasi

   Tahap ini merupakan tahap dimana peneliti mengimplementasikan system yang telah dibuat ke sekolah yang telah ditentukan.

F. Metode Pengujian

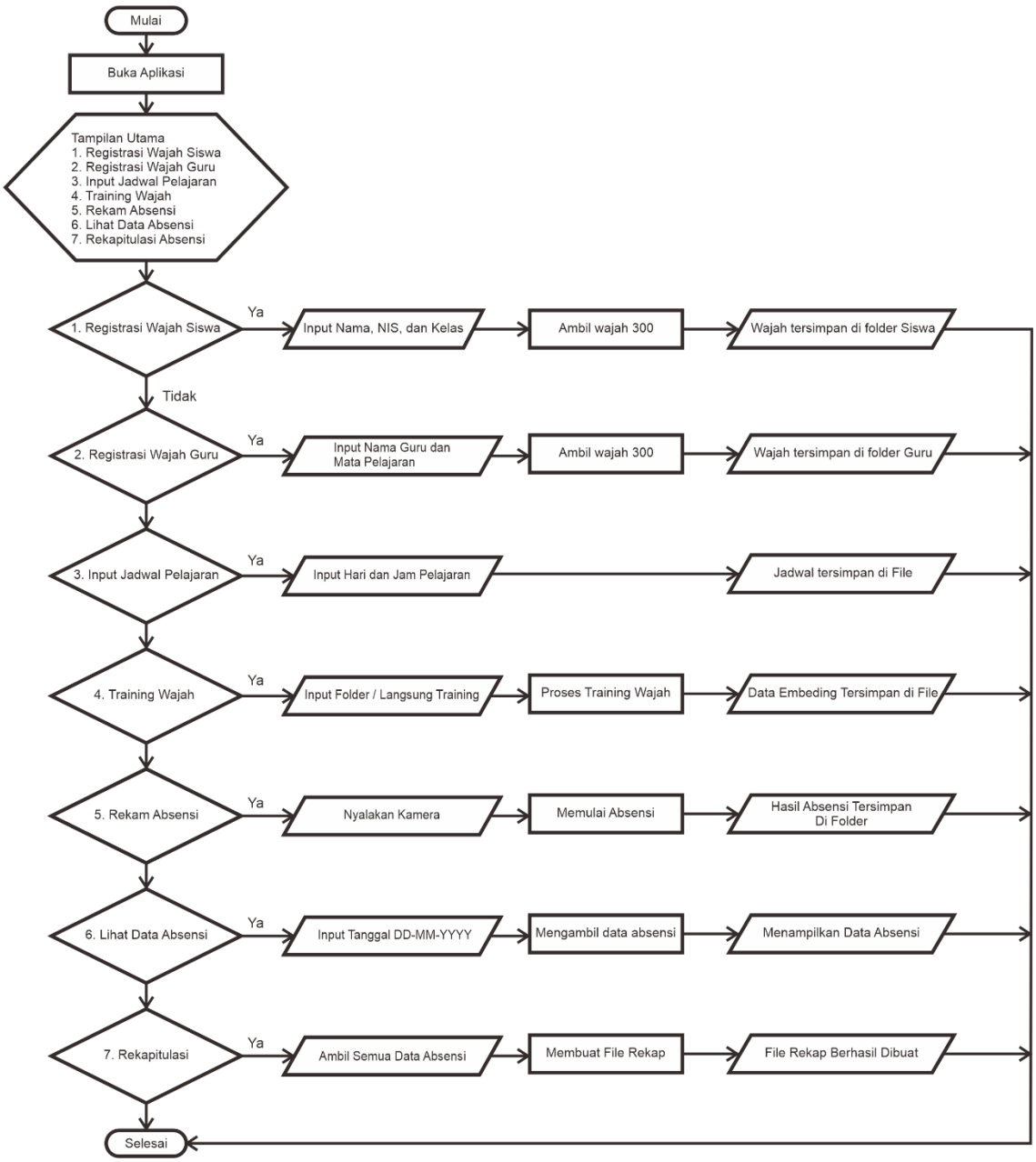
Setelah aplikasi telah dibuat maka dilakukan pula pengujian pada aplikasi tersebut. Dalam penelitian ini digunakan 2 (Dua) metode untuk melakukan pengujian pada sistem aplikasi yaitu blackbox testing dan whitebox testing.

BAB IV

HASIL DAN PEMBAHASA

A. Diagram Alir

Flowchart Aplikasi



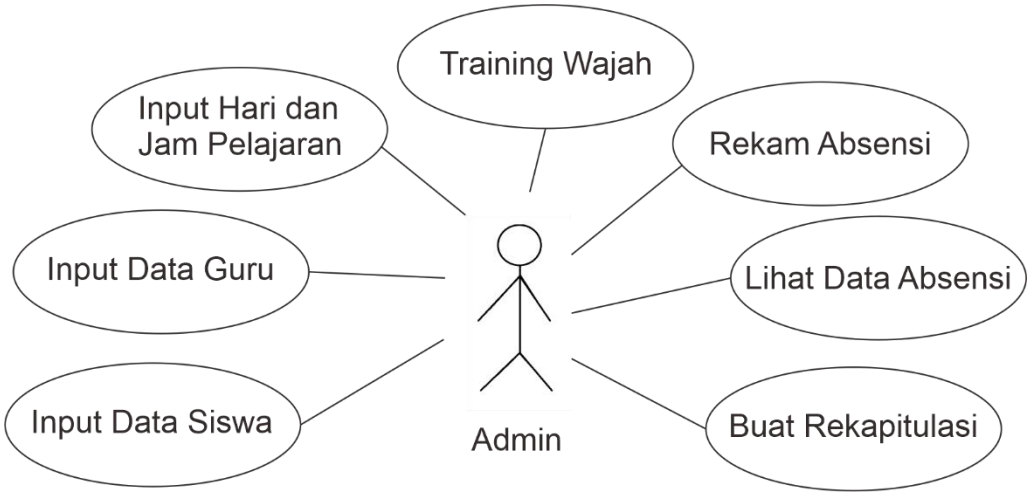
Gambar 4.1 *Flowchart* Aplikasi

Pengembangan sistem absensi ini dimulai dengan membuka aplikasi yang menampilkan menu utama berisi beberapa fitur, yaitu: registrasi wajah siswa, registrasi wajah guru, input jadwal pelajaran, training wajah, rekam absensi, lihat data absensi, dan rekapitulasi absensi. Apabila pengguna memilih registrasi wajah siswa, maka pengguna akan diminta untuk menginput nama, NIS, dan kelas siswa, kemudian sistem akan mengambil 300 gambar wajah yang disimpan secara otomatis ke dalam folder khusus siswa. Jika pengguna memilih registrasi wajah guru, maka nama guru dan mata pelajaran akan diinput, dilanjutkan dengan pengambilan 300 wajah yang kemudian disimpan ke dalam folder guru. Selanjutnya, pada fitur input jadwal pelajaran, pengguna akan mengisi hari dan jam pelajaran yang kemudian disimpan dalam sebuah file jadwal.

Proses berikutnya adalah *training* wajah, yang memungkinkan pengguna memilih folder wajah atau langsung memulai proses *training*. Sistem kemudian memproses data wajah dan menyimpan hasil embedding ke dalam file. Setelah data wajah terlatih, proses rekam absensi dapat dilakukan dengan menyalakan kamera dan memulai sistem absensi berbasis real-time. Hasil absensi otomatis akan disimpan dalam folder. Untuk melihat data absensi, pengguna cukup menginput tanggal dalam format DD-MM-YYYY, maka sistem akan mengambil dan menampilkan data absensi yang sesuai. Terakhir, sistem menyediakan fitur rekapitulasi dengan mengambil seluruh data absensi dan membuat file rekap secara otomatis. Setelah seluruh proses selesai, maka sistem dinyatakan selesai menjalankan alur absensi dengan teknologi face recognition berbasis YOLO secara menyeluruh dan sistematis.

B. Analisis Sistem Yang Diusulkan

1. Use Case Diagram



Gambar 4.2 Use case sistem yang diusulkan

Use case diagram digunakan untuk melihat hubungan antar actor dan sistem. Mendeskripsikan interaksi antara keduanya.

Tabel 4.1 Penjelasan use case diagram

Nama Use case	Deskripsi *Use Case*
Input Data Siswa	Admin dapat memasukkan data identitas siswa, termasuk nama, NIS, kelas, dan data wajah yang akan digunakan dalam proses pengenalan wajah saat absensi.
Input Data Guru	Admin bertanggung jawab untuk memasukkan data guru, seperti nama guru dan mata pelajaran yang diampu, yang nantinya dikaitkan dengan jadwal pelajaran.
Input Hari dan Jam Pelajaran	Admin dapat mengatur jadwal pelajaran dengan menentukan hari dan jam, serta mengaitkannya dengan guru dan mata pelajaran yang sesuai.
Training Wajah	Admin menjalankan proses pelatihan wajah, yaitu melatih model AI dengan dataset wajah siswa agar sistem mampu mengenali wajah secara akurat saat proses absensi.

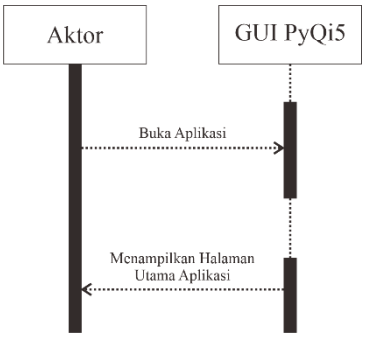
Lanjutan Tabel 4.1

Nama Use case	Deskripsi *Use Case*
Rekam Absensi	Admin dapat memulai proses absensi dengan merekam kehadiran siswa berdasarkan hasil pengenalan wajah secara otomatis dan real-time.
Lihat Data Absensi	Admin dapat mengakses data absensi yang telah direkam, baik untuk memantau kehadiran siswa maupun untuk keperluan evaluasi.
Buat Rekapitulasi	Admin dapat membuat laporan rekap absensi yang menampilkan data statistik kehadiran siswa secara keseluruhan, yang bisa diekspor atau disimpan dalam bentuk Excell

2. Sequence Diagram

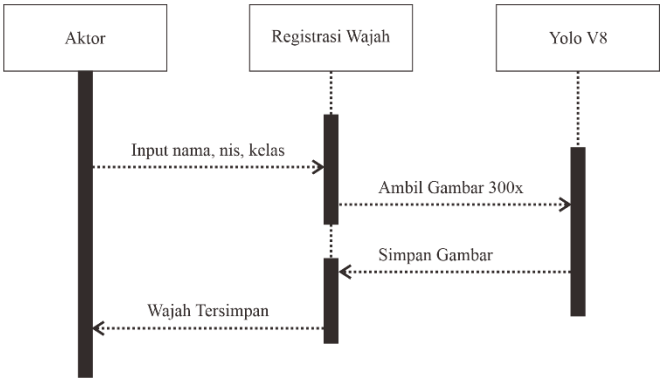
Sequence diagram menggambarkan interaksi antara objek dalam suatu sistem berdasarkan urutan waktu, membantu memahami alur komunikasi dalam sistem.

a. Sequence diagram membuka aplikasi



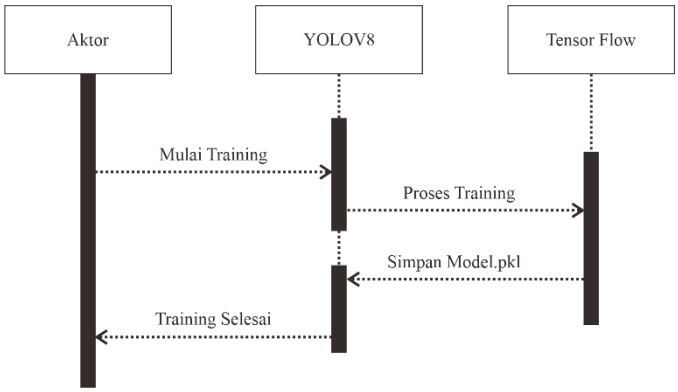
Gambar 4.3 Sequence diagram membuka aplikasi

b. Sequence Diagram Input wajah



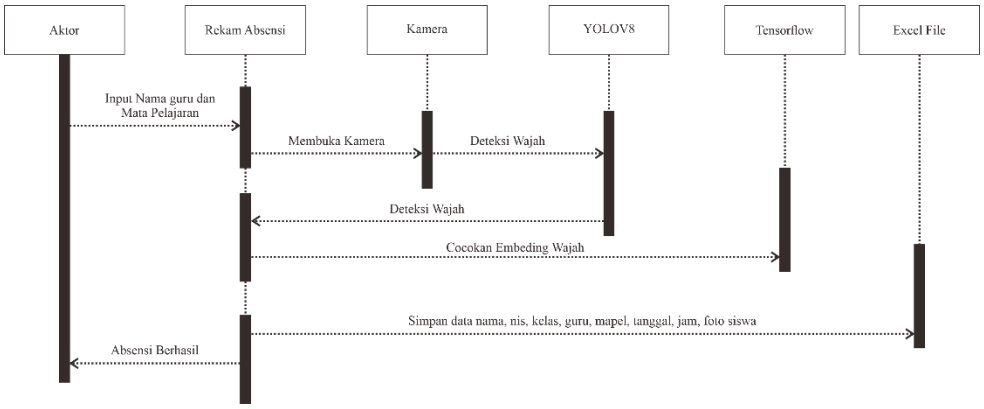
Gambar 4.4 Sequence diagram Input Wajah

c. Sequence Diragram Training Wajah



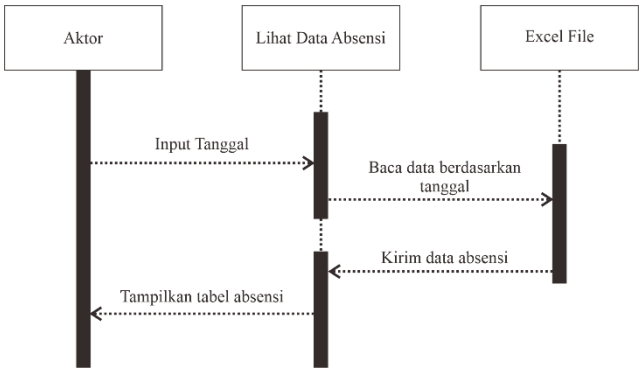
Gambar 4.5 Sequence Diagram Training Wajah

d. Sequence Diagram Rekam Absensi



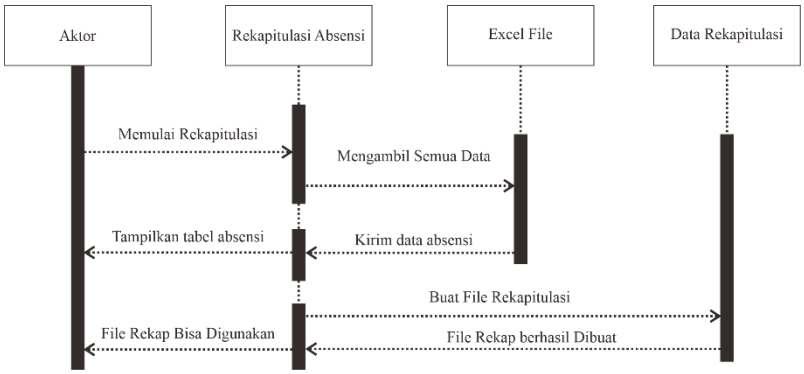
Gambar 4.6 Sequence Diagram Rekam Absensi

e. Sequence Diagram Lihat Data Absensi



Gambar 4.7 Sequence Diagram Lihat Data Absensi

f. Sequence Diagram Rekapitulasi Absensi



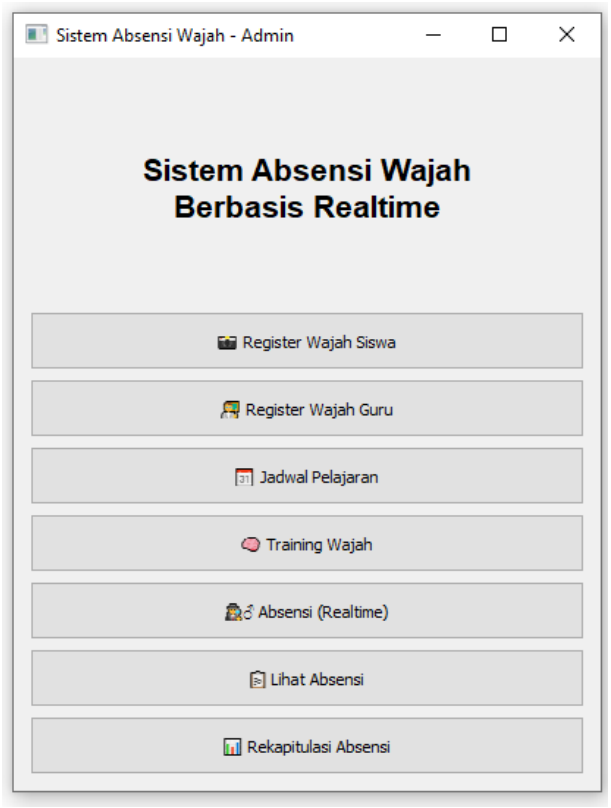
Gambar 4.8 Sequence Diagram Rekapitulasi Absensi

C. Hasil Implementasi Sistem

Penelitian ini menghasilkan sebuah aplikasi sistem absensi berbasis pengenalan wajah (face recognition) yang bekerja secara real-time menggunakan YOLOv8 untuk deteksi wajah, TensorFlow untuk pemrosesan embedding wajah, serta PyQt5 sebagai antarmuka pengguna. Data absensi disimpan dalam format Excel untuk kemudahan manajemen dan integrasi.

Adapun tampilan, fitur dan hasil dari pengumpulan data yang saya dapatkan ketika meneliti adalah sebagai berikut :

1. Tampilan Utama



Gambar 4.9 Tampilan Utama Aplikasi

```
from PyQt5.QtWidgets import QWidget, QVBoxLayout,
QPushButton, QLabel
from PyQt5.QtGui import QFont
from PyQt5.QtCore import Qt

from ui.register_siswa import RegisterSiswaWindow
from ui.register_guru import RegisterGuruWindow
from ui.jadwal_window import JadwalWindow
from ui.training_window import TrainingWindow
from ui.absensi_window import AbsensiWindow
from ui.lihat_absensi import LihatAbsensiWindow
from ui.rekap_window import RekapWindow

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Sistem Absensi Wajah -
Admin")
        self.setGeometry(100, 100, 400, 500)

        layout = QVBoxLayout()

        title = QLabel("Sistem Absensi Wajah\nBerbasis
Realtime")
        title.setAlignment(Qt.AlignCenter)
        title.setFont(QFont("Arial", 16, QFont.Bold))
        layout.addWidget(title)
```

```
        buttons = [
            ("📸 Register Wajah Siswa",
RegisterSiswaWindow),
            ("👨‍🏫 Register Wajah Guru",
RegisterGuruWindow),
            ("📅 Jadwal Pelajaran", JadwalWindow),
            ("🧠 Training Wajah", TrainingWindow),
            ("🕵️ Absensi (Realtime)", AbsensiWindow),
            ("📋 Lihat Absensi", LihatAbsensiWindow),
            ("📊 Rekapitulasi Absensi", RekapWindow),
        ]

        for text, window_class in buttons:
            btn = QPushButton(text)
            btn.setMinimumHeight(40)
            btn.clicked.connect(lambda checked,
cls=window_class: self.buka_window(cls))
            layout.addWidget(btn)

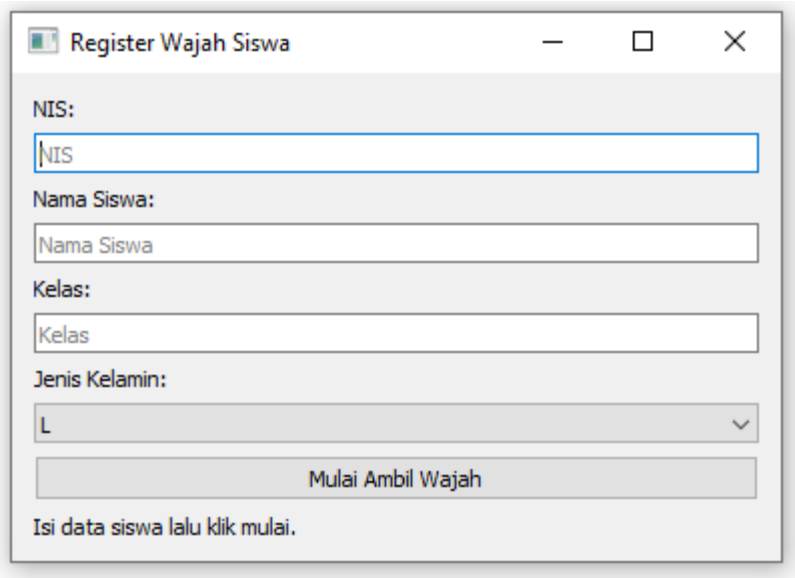
        self.setLayout(layout)

    def buka_window(self, window_class):
        self.window = window_class()
        self.window.show()
```

Kode python diatas merupakan antarmuka utama aplikasi absensi wajah berbasis PyQt5 yang menyediakan tombol navigasi untuk membuka fitur pada aplikasi tersebut.

Tampilan utama aplikasi Sistem Absensi Wajah Berbasis Realtime dirancang dengan antarmuka yang sederhana, informatif, dan mudah digunakan oleh admin. Pada bagian atas, terdapat judul aplikasi yang ditampilkan dalam huruf tebal dan berukuran besar, yaitu "Sistem Absensi Wajah Berbasis Realtime", yang secara langsung memberikan gambaran bahwa sistem ini memanfaatkan teknologi pengenalan wajah untuk mencatat kehadiran secara otomatis dan realtime. Di bawah judul tersebut, terdapat tujuh tombol menu utama yang disusun secara vertikal dan rapi, masing-masing dilengkapi dengan ikon yang sesuai untuk memperjelas fungsi dari setiap fitur.

2. Registrasi Wajah Siswa



Gambar 4.10 Registrasi Wajah Siswa

```
import os
import cv2
from PyQt5.QtWidgets import QWidget, QLabel, QLineEdit,
QPushButton, QVBoxLayout, QMessageBox
from PyQt5.QtCore import QTimer
from PyQt5.QtGui import QIntValidator
from logic.face_utils import detect_faces, crop_face

SAVE_DIR = "data/siswa"
TOTAL_IMAGES = 300
CAMERA_INDEX = 1  # Ubah ini kalau kamera eksternal tidak
terbaca

class RegisterSiswaWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Register Wajah Siswa")
        self.image_count = 0
        self.capture = None

        # Input
        self.nama_input = QLineEdit()
        self.nama_input.setPlaceholderText("Nama Siswa")

        self.nis_input = QLineEdit()
        self.nis_input.setPlaceholderText("NIS")
        self.nis_input.setValidator(QIntValidator())  # Hanya
izinkan angka

        self.kelas_input = QLineEdit()
        self.kelas_input.setPlaceholderText("Kelas")

        self.start_button = QPushButton("Mulai Ambil Wajah")
        self.start_button.clicked.connect(self.start_capture)

        self.info_label = QLabel("Isi data siswa lalu klik
mulai.")
```

```
        self.layout = QVBoxLayout()
        self.layout.addWidget(self.nama_input)
        self.layout.addWidget(self.nis_input)
        self.layout.addWidget(self.kelas_input)
        self.layout.addWidget(self.start_button)
        self.layout.addWidget(self.info_label)

        self.setLayout(self.layout)

        # Timer
        self.timer = QTimer()
        self.timer.timeout.connect(self.update_frame)

    def start_capture(self):
        nama = self.nama_input.text().strip()
        nis = self.nis_input.text().strip()
        kelas = self.kelas_input.text().strip()

        if not nama or not nis or not kelas:
            QMessageBox.warning(self, "Peringatan", "Semua kolom
harus diisi!")
            return

        self.folder_name = f"{nama}_{nis}_{kelas}".replace(" ",
"_")
        self.save_path = os.path.join(SAVE_DIR,
self.folder_name)
        os.makedirs(self.save_path, exist_ok=True)

        self.capture = cv2.VideoCapture(CAMERA_INDEX)
        self.image_count = 0
        self.timer.start(30)
        self.info_label.setText("Mengambil gambar... (Tekan 'q'
untuk berhenti)")

    def update_frame(self):
        ret, frame = self.capture.read()
        if not ret:
            self.info_label.setText("Gagal mengakses kamera.")
            return

        boxes = detect_faces(frame)
        for box in boxes:
            x1, y1, x2, y2 = box
            # Tambahkan bounding box
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255,
0), 2)

            # Tambahkan teks jumlah gambar
            cv2.putText(frame, f"Gambar:
{self.image_count}/{TOTAL_IMAGES}", (x1, y1 - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255,
0), 2)

            # Simpan gambar wajah
            if self.image_count < TOTAL_IMAGES:
                face = crop_face(frame, box)
                img_path = os.path.join(self.save_path,
f"{self.image_count + 1}.jpg")
                cv2.imwrite(img_path, face)
                self.image_count += 1
```

```
            else:
                self.timer.stop()
                self.capture.release()
                self.info_label.setText("Pengambilan wajah
selesai.")
                QMessageBox.information(self, "Selesai",
f"{TOTAL_IMAGES} wajah telah disimpan.")
                cv2.destroyAllWindows()
                return

        # Tampilkan frame dan periksa tombol q
        cv2.imshow("Pendaftaran Wajah Siswa", frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            self.timer.stop()
            self.capture.release()
            self.info_label.setText("Pengambilan dihentikan oleh
pengguna.")
            QMessageBox.information(self, "Dihentikan",
f"{self.image_count} gambar telah disimpan.")
            cv2.destroyAllWindows()

    def closeEvent(self, event):
        self.timer.stop()
        if self.capture:
            self.capture.release()
        cv2.destroyAllWindows()
```

Kode python diatas merupakan jendela registrasi wajah siswa menggunakan PyQt5 yang memungkinkan pengguna mengisi data siswa dan mengambil 300 gambar wajah secara otomatis dengan deteksi YOLOv8 untuk disimpan sebagai dataset.

Tampilan antarmuka dari fitur "Register Wajah Siswa" dalam aplikasi sistem absensi wajah. Tampilan ini merupakan formulir pendaftaran wajah siswa, yang digunakan untuk menginput data identitas sebelum pengambilan gambar wajah dilakukan.

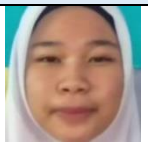
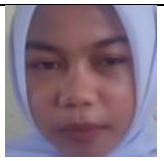
pengguna (siswa) mendaftarkan wajah dengan memasukkan nama, NIS, dan kelas. Gambar wajah disimpan dalam direktori tertentu. Data yang saya kumpulkan yaitu 14 siswa dan masing-masing foto siswa sebanyak 300 gambar yang berarti 4.200 gambar, dengan data tabel sebagai berikut :

Tabel 4.2 Data Wajah Siswa

NIS	Nama	Kelas	Foto

241388	Adhe Andriansyah	X TKJ	
241389	Andi Nurmayani	X TKJ	
241390	Farel B	X TKJ	
241392	Khaerunnisaa Biqalbi	X TKJ	
241393	Muh. Pasha Syaputra	X TKJ	

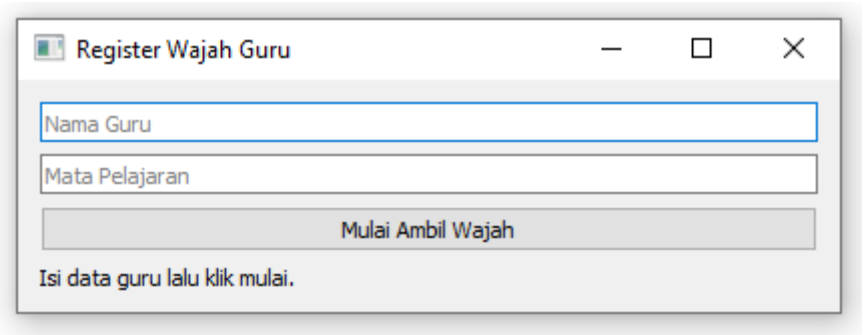
Lanjutan Tabel 4.2

NIS	Nama	Kelas	Foto
241395	Nuraniza Ramadani	X TKJ	
241397	Selvi	X TKJ	
241398	Syaiful Rahmat Ilyas	X TKJ	
241399	Tiara Husnul Khatimah D.	X TKJ	
241398	Amran Serman	X TAV	
241399	Muh. Fausan	X TAV	
241400	Muh. Nasrun	X TAV	
241401	Muhammad Yusran	X TAV	

Lanjutan Tabel 4.2

NIS	Nama	Kelas	Foto
241403	Muh. Fadil	X TAV	

3. Registrasi Wajah Guru



Gambar 4.11 Registrasi Wajah Guru

```
import os
import cv2
from PyQt5.QtWidgets import QWidget, QLabel, QLineEdit,
QPushButton, QVBoxLayout, QMessageBox
from PyQt5.QtCore import QTimer
from logic.face_utils import detect_faces, crop_face,
draw_bounding_box

SAVE_DIR = "data/guru"
TOTAL_IMAGES = 300
CAMERA_INDEX = 1  # Ubah jika kamera eksternal tidak terbaca

class RegisterGuruWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Register Wajah Guru")
        self.image_count = 0
        self.capture = None

        # Input
        self.nama_input = QLineEdit()
        self.nama_input.setPlaceholderText("Nama Guru")
        self.mapel_input = QLineEdit()
        self.mapel_input.setPlaceholderText("Mata Pelajaran")

        self.start_button = QPushButton("Mulai Ambil Wajah")
        self.start_button.clicked.connect(self.start_capture)

          self.info_label = QLabel("Isi data guru lalu klik
mulai.")
        self.layout = QVBoxLayout()
        self.layout.addWidget(self.nama_input)
        self.layout.addWidget(self.mapel_input)
        self.layout.addWidget(self.start_button)
```

```
        self.layout.addWidget(self.info_label)

        self.setLayout(self.layout)

        # Timer
        self.timer = QTimer()
        self.timer.timeout.connect(self.update_frame)

    def start_capture(self):
        nama = self.nama_input.text().strip()
        mapel = self.mapel_input.text().strip()

        if not nama or not mapel:
              QMessageBox.warning(self, "Peringatan", "Semua
kolom harus diisi!")
            return

       self.folder_name = f"{nama}_{mapel}".replace(" ", "_")
                 self.save_path   =   os.path.join(SAVE_DIR,
self.folder_name)
        os.makedirs(self.save_path, exist_ok=True)

        self.capture = cv2.VideoCapture(CAMERA_INDEX)
        self.image_count = 0
        self.timer.start(30)
       self.info_label.setText("Mengambil gambar... Tekan 'q'
untuk berhenti.")

    def update_frame(self):
        ret, frame = self.capture.read()
        if not ret:
           self.info_label.setText("Gagal mengakses kamera.")
            return

        boxes = detect_faces(frame)
        for box in boxes:
            draw_bounding_box(frame, box)
            x, y, w, h = box
             # Tambahkan teks jumlah gambar di atas bounding
box
                                          cv2.putText(frame,
f"{self.image_count}/{TOTAL_IMAGES}", (x, y - 10),
                           cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0,
255, 0), 2)

            if self.image_count < TOTAL_IMAGES:
                face = crop_face(frame, box)
                     img_path  =  os.path.join(self.save_path,
f"{self.image_count + 1}.jpg")
                cv2.imwrite(img_path, face)
                self.image_count += 1
            else:
                self.timer.stop()
                self.capture.release()
                   self.info_label.setText("Pengambilan wajah
selesai.")
                     QMessageBox.information(self, "Selesai",
f"{TOTAL_IMAGES} wajah telah disimpan.")
                cv2.destroyAllWindows()
                return
```

```
        # Tampilkan frame
        cv2.imshow("Pendaftaran Wajah Guru", frame)

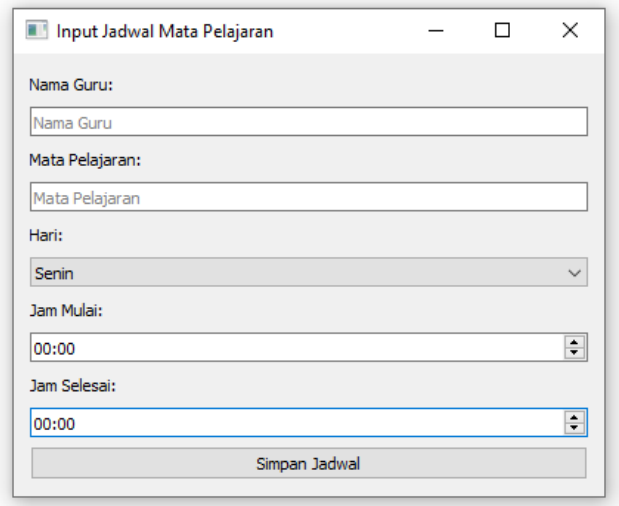
        # Deteksi tombol 'q' untuk menghentikan secara manual
        if cv2.waitKey(1) & 0xFF == ord('q'):
            self.timer.stop()
            self.capture.release()
            self.info_label.setText(f"Pengambilan dihentikan.
{self.image_count} gambar disimpan.")
                QMessageBox.information(self, "Dihentikan",
f"Pengambilan dihentikan.\n{self.image_count} gambar telah
disimpan.")
            cv2.destroyAllWindows()

    def closeEvent(self, event):
        self.timer.stop()
        if self.capture:
            self.capture.release()
        cv2.destroyAllWindows()
```

Kode python diatas merupakan jendela registrasi wajah Guru menggunakan PyQt5 yang memungkinkan pengguna mengisi data guru dan mengambil 300 gambar wajah secara otomatis dengan deteksi YOLOv8 untuk disimpan sebagai dataset.

Menu yang ditampilkan pada gambar merupakan antarmuka untuk fitur Register Wajah Guru dalam sistem absensi berbasis face recognition. Pada tampilan ini, pengguna diminta untuk mengisi dua input penting, yaitu Nama Guru dan Mata Pelajaran yang diajarkan. Kedua kolom ini wajib diisi sebelum proses pengambilan wajah dapat dimulai. Di bawah kolom input terdapat tombol bertuliskan "Mulai Ambil Wajah", yang akan diaktifkan setelah data guru lengkap diinput. Tombol ini berfungsi untuk memulai proses perekaman wajah guru menggunakan kamera. Di bagian bawah jendela, terdapat petunjuk berupa teks: *"Isi data guru lalu klik mulai."* yang memberikan instruksi singkat kepada pengguna untuk mengisi data terlebih dahulu sebelum memulai proses. Tampilan ini dirancang dengan sederhana dan jelas untuk mempermudah pengguna dalam melakukan registrasi wajah guru ke dalam sistem.

4. Jadwal Pelajaran



Gambar 4.12 Input Jadwal Mata Pelajaran

```
import os
import pandas as pd
from PyQt5.QtWidgets import QWidget, QLabel, QLineEdit, QPushButton,
QVBoxLayout, QComboBox, QMessageBox, QTimeEdit
from PyQt5.QtCore import QTime

JADWAL_PATH = "data/jadwal.xlsx"

class JadwalWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Input Jadwal Mata Pelajaran")
        self.setGeometry(200, 200, 400, 300)

        # Input
        self.guru_input = QLineEdit()
        self.guru_input.setPlaceholderText("Nama Guru")

        self.mapel_input = QLineEdit()
        self.mapel_input.setPlaceholderText("Mata Pelajaran")

        self.hari_combo = QComboBox()
            self.hari_combo.addItems(["Senin",  "Selasa",  "Rabu",
"Kamis", "Jumat", "Sabtu"])

        self.jam_mulai = QTimeEdit()
        self.jam_mulai.setDisplayFormat("HH:mm")
        self.jam_mulai.setTime(QTime.currentTime())

        self.jam_selesai = QTimeEdit()
        self.jam_selesai.setDisplayFormat("HH:mm")
        self.jam_selesai.setTime(QTime.currentTime().addSecs(3600))

        self.submit_button = QPushButton("Simpan Jadwal")
        self.submit_button.clicked.connect(self.simpan_jadwal)

        # Layout
        layout = QVBoxLayout()
        layout.addWidget(QLabel("Nama Guru:"))
```

```
        layout.addWidget(self.guru_input)
        layout.addWidget(QLabel("Mata Pelajaran:"))
        layout.addWidget(self.mapel_input)
        layout.addWidget(QLabel("Hari:"))
        layout.addWidget(self.hari_combo)
        layout.addWidget(QLabel("Jam Mulai:"))
        layout.addWidget(self.jam_mulai)
        layout.addWidget(QLabel("Jam Selesai:"))
        layout.addWidget(self.jam_selesai)
        layout.addWidget(self.submit_button)

        self.setLayout(layout)

    def simpan_jadwal(self):
        guru = self.guru_input.text().strip()
        mapel = self.mapel_input.text().strip()
        hari = self.hari_combo.currentText()
        jam_mulai = self.jam_mulai.time().toString("HH:mm")
        jam_selesai = self.jam_selesai.time().toString("HH:mm")

        if not guru or not mapel:
              QMessageBox.warning(self, "Peringatan", "Semua kolom
harus diisi!")
            return

        data = {
            "Nama Guru": [guru],
            "Mata Pelajaran": [mapel],
            "Hari": [hari],
            "Jam Mulai": [jam_mulai],
            "Jam Selesai": [jam_selesai],
        }

        df_new = pd.DataFrame(data)

        if os.path.exists(JADWAL_PATH):
            df_existing = pd.read_excel(JADWAL_PATH)
                  df_combined  =  pd.concat([df_existing,  df_new],
ignore_index=True)
        else:
            df_combined = df_new

        df_combined.to_excel(JADWAL_PATH, index=False)
         QMessageBox.information(self, "Berhasil", "Jadwal berhasil
disimpan.")
        self.close()
```

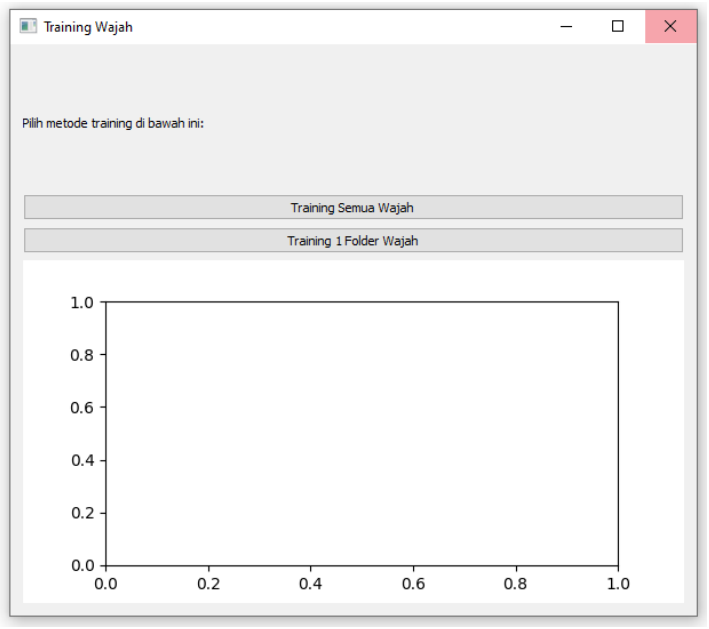
Script ini adalah antarmuka PyQt5 untuk menampilkan input data mata pelajaran yang akan digunakan untuk absensi dan akan tersimpan di file Excel.

Gambar 4.13 Database Jadwal Pelajaran

	A	B	C	D	E
1	**Nama Guru**	**Mata Pelajaran**	**Hari**	**Jam Mulai**	**Jam Selesai**
2	Nurhidayani, S.Kom	Dasar - Dasar Program Keahlian TJKT	Senin	08:00	12:00
3	Andi Rifaldy Wijaya, S.Pd.	Informatika	Senin	12:30	14:50
4	Nurhidayani, S.Kom	Dasar - Dasar Program Keahlian TJKT	Selasa	07:15	11:20
5	Suci Reskiana Putri, S.Pd.	Akidah Akhlak	Selasa	11:20	13:05
6	Marlia Amalia Umar, S.Pd.	Bahasa Indonesia	Selasa	13:05	15:25
7	Abdul Mannan S.Pd.,M.Pd.	PJOK	Rabu	07:15	09:00
8	Wigra Muftiah	Kemuhammadiyahan	Rabu	09:00	10:45
9	Wigra Muftiah	Bahasa Arab	Rabu	10:45	12:00
10	Wigra Muftiah	Tarikh	Rabu	12:00	13:40
11	Yeyen Retni Nasa, S.Pd.	Seni Budaya	Rabu	13:40	14:50
12	Suci Reskiana Putri, S.Pd.	Pendidikan Pancasila	Kamis	07:15	08:25
13	Wigra Muftiah	Fiqih	Kamis	08:25	09:35
14	Andi Rifaldy Wijaya, S.Pd.	Matematika	Kamis	10:10	13:05
15	Yeyen Retni Nasa, S.Pd.	Bahasa Inggris	Kamis	13:05	15:25
16	Yeyen Retni Nasa, S.Pd.	Sejarah	Jumat	07:15	08:25
17	Mariana B, S.Pd.	IPAS	Jumat	08:25	13:35
18	Suci Reskiana Putri, S.Pd.	Al qur'an Hadist	Jumat	13:35	14:50

Menu yang ditampilkan pada gambar merupakan antarmuka Input Jadwal Mata Pelajaran, yang berfungsi untuk mencatat jadwal mengajar setiap guru dalam sistem absensi berbasis face recognition. Di bagian atas terdapat dua kolom input teks untuk mengisi Nama Guru dan Mata Pelajaran, yang digunakan untuk mengidentifikasi data guru terkait. Selanjutnya, terdapat dropdown menu Hari, yang memungkinkan pengguna memilih hari mengajar dari Senin hingga Minggu. Di bawahnya, terdapat dua pilihan waktu yaitu Jam Mulai dan Jam Selesai, yang digunakan untuk menentukan durasi jadwal mengajar. Setelah semua data diisi dengan benar, pengguna dapat menekan tombol "Simpan Jadwal" untuk menyimpan data ke sistem. Antarmuka ini dirancang untuk memastikan bahwa informasi jadwal setiap guru dapat tercatat dengan rapi dan akurat, serta terintegrasi dalam proses absensi otomatis berdasarkan waktu dan mata pelajaran yang sedang berlangsung.

5. Training Wajah



Gambar 4.14 Training Wajah

```
from PyQt5.QtWidgets import QWidget, QPushButton, QLabel,
QVBoxLayout, QMessageBox, QFileDialog
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
FigureCanvas
from matplotlib.figure import Figure
from logic.training_utils import train_embeddings

class TrainingWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Training Wajah")
        self.setGeometry(200, 200, 600, 500)

        self.info_label = QLabel("Pilih metode training di bawah
ini:")

        self.train_all_button = QPushButton("Training Semua Wajah")
        self.train_all_button.clicked.connect(self.training_semua)

        self.train_one_button = QPushButton("Training 1 Folder
Wajah")
        self.train_one_button.clicked.connect(self.training_satu_fo
lder)

        # Canvas untuk plotting matplotlib
        self.canvas = FigureCanvas(Figure(figsize=(5, 3)))
        self.ax = self.canvas.figure.subplots()

        layout = QVBoxLayout()
        layout.addWidget(self.info_label)
        layout.addWidget(self.train_all_button)
        layout.addWidget(self.train_one_button)
        layout.addWidget(self.canvas)

        self.setLayout(layout)
```

```
    def training_semua(self):
        print("=== Mulai Training Semua Wajah ===")
        self.info_label.setText("Proses training semua wajah sedang
berjalan...")

        jumlah_siswa, jumlah_guru, history = train_embeddings()

        print(f"Training selesai: {jumlah_siswa} wajah siswa,
{jumlah_guru} wajah guru")

        QMessageBox.information(
            self,
            "Training Selesai",
            f"Training semua selesai!\n\nSiswa: {jumlah_siswa}
wajah\nGuru: {jumlah_guru} wajah"
        )

        if history:
            print("Menampilkan grafik akurasi dan loss...")
            self.plot_history(history)
        else:
            print("Tidak ada riwayat training (history tidak
tersedia)")

        self.info_label.setText("Pilih metode training di bawah
ini:")

    def training_satu_folder(self):
        folder_path = QFileDialog.getExistingDirectory(self, "Pilih
Folder Wajah")
        if folder_path:
            print(f"=== Mulai Training Folder: {folder_path} ===")
            self.info_label.setText(f"Proses training folder:
{folder_path}")

            jumlah, history =
train_embeddings(folder_path=folder_path)

            print(f"Training folder selesai: {jumlah} wajah")

            QMessageBox.information(
                self,
                "Training Selesai",
                f"Training folder selesai!\n\nJumlah wajah:
{jumlah}"
            )

            if history:
                print("Menampilkan grafik akurasi dan loss...")
                self.plot_history(history)
            else:
                print("Tidak ada riwayat training (history tidak
tersedia)")

            self.info_label.setText("Pilih metode training di bawah
ini:")

    def plot_history(self, history):
        self.ax.clear()

        if 'accuracy' in history.history:
```

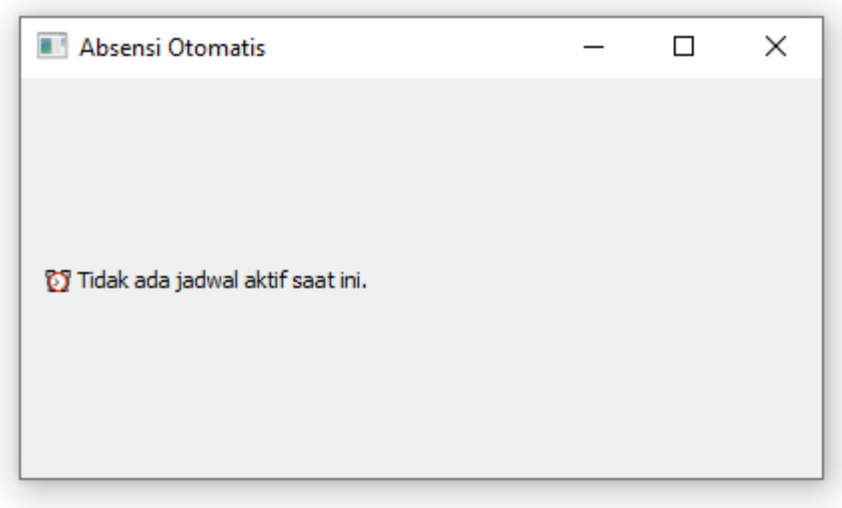
```
            self.ax.plot(history.history['accuracy'], label='Train
Accuracy')
        if 'val_accuracy' in history.history:
            self.ax.plot(history.history['val_accuracy'],
label='Val Accuracy')
        if 'loss' in history.history:
            self.ax.plot(history.history['loss'], label='Train
Loss')
        if 'val_loss' in history.history:
            self.ax.plot(history.history['val_loss'], label='Val
Loss')

        self.ax.set_title("Training Accuracy & Loss")
        self.ax.set_xlabel("Epoch")
        self.ax.set_ylabel("Value")
        self.ax.legend()
        self.canvas.draw()
```

Script ini merupakan antarmuka pelatihan wajah dengan PyQt5 yang mengekstraksi embedding wajah dari dataset menggunakan DeepFace dan menyimpannya beserta label ke dalam file model.pkl untuk digunakan dalam proses pengenalan wajah.

Menu yang ditampilkan pada gambar merupakan antarmuka Training Wajah, yang berfungsi untuk melatih sistem dalam mengenali wajah-wajah yang telah direkam sebelumnya, baik dari guru maupun siswa. Pada bagian atas terdapat petunjuk yang berbunyi *"Pilih metode training di bawah ini:"* yang mengarahkan pengguna untuk memilih salah satu dari dua metode pelatihan yang tersedia. Tombol pertama bertuliskan "Training Semua Wajah", yang akan memproses seluruh data wajah yang tersimpan dalam sistem secara otomatis. Sementara tombol kedua yaitu "Training 1 Folder Wajah" memungkinkan pengguna untuk memilih dan melatih wajah dari satu folder tertentu saja, sesuai kebutuhan. Di bagian bawah, terdapat area grafik yang digunakan untuk menampilkan progres, visualisasi hasil, atau status dari proses training. Menu ini memiliki peran penting dalam memastikan bahwa data wajah yang telah direkam dapat diproses dan dikenali secara optimal oleh sistem absensi berbasis face recognition.

6. Rekam Absensi



Gambar 4.15 Absensi Wajah

```
from PyQt5.QtWidgets import QWidget, QPushButton, QLabel,
QVBoxLayout, QMessageBox, QFileDialog
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
FigureCanvas
from matplotlib.figure import Figure
from logic.training_utils import train_embeddings

class TrainingWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Training Wajah")
        self.setGeometry(200, 200, 600, 500)

        self.info_label = QLabel("Pilih metode training di bawah
ini:")

        self.train_all_button = QPushButton("Training Semua Wajah")
        self.train_all_button.clicked.connect(self.training_semua)

        self.train_one_button = QPushButton("Training 1 Folder
Wajah")
        self.train_one_button.clicked.connect(self.training_satu_fo
lder)

        # Canvas untuk plotting matplotlib
        self.canvas = FigureCanvas(Figure(figsize=(5, 3)))
        self.ax = self.canvas.figure.subplots()

        layout = QVBoxLayout()
        layout.addWidget(self.info_label)
        layout.addWidget(self.train_all_button)
        layout.addWidget(self.train_one_button)
        layout.addWidget(self.canvas)

        self.setLayout(layout)

    def training_semua(self):
        print("=== Mulai Training Semua Wajah ===")
        self.info_label.setText("Proses training semua wajah sedang
berjalan...")
```

```
        jumlah_siswa, jumlah_guru, history = train_embeddings()

        print(f"Training selesai: {jumlah_siswa} wajah siswa,
{jumlah_guru} wajah guru")

        QMessageBox.information(
            self,
            "Training Selesai",
            f"Training semua selesai!\n\nSiswa: {jumlah_siswa}
wajah\nGuru: {jumlah_guru} wajah"
        )

        if history:
            print("Menampilkan grafik akurasi dan loss...")
            self.plot_history(history)
        else:
            print("Tidak ada riwayat training (history tidak
tersedia)")

        self.info_label.setText("Pilih metode training di bawah
ini:")

    def training_satu_folder(self):
        folder_path = QFileDialog.getExistingDirectory(self, "Pilih
Folder Wajah")
        if folder_path:
            print(f"=== Mulai Training Folder: {folder_path} ===")
            self.info_label.setText(f"Proses training folder:
{folder_path}")

            jumlah, history =
train_embeddings(folder_path=folder_path)

            print(f"Training folder selesai: {jumlah} wajah")

            QMessageBox.information(
                self,
                "Training Selesai",
                f"Training folder selesai!\n\nJumlah wajah:
{jumlah}"
            )

            if history:
                print("Menampilkan grafik akurasi dan loss...")
                self.plot_history(history)
            else:
                print("Tidak ada riwayat training (history tidak
tersedia)")

            self.info_label.setText("Pilih metode training di bawah
ini:")

    def plot_history(self, history):
        self.ax.clear()

        if 'accuracy' in history.history:
            self.ax.plot(history.history['accuracy'], label='Train
Accuracy')
        if 'val_accuracy' in history.history:
```

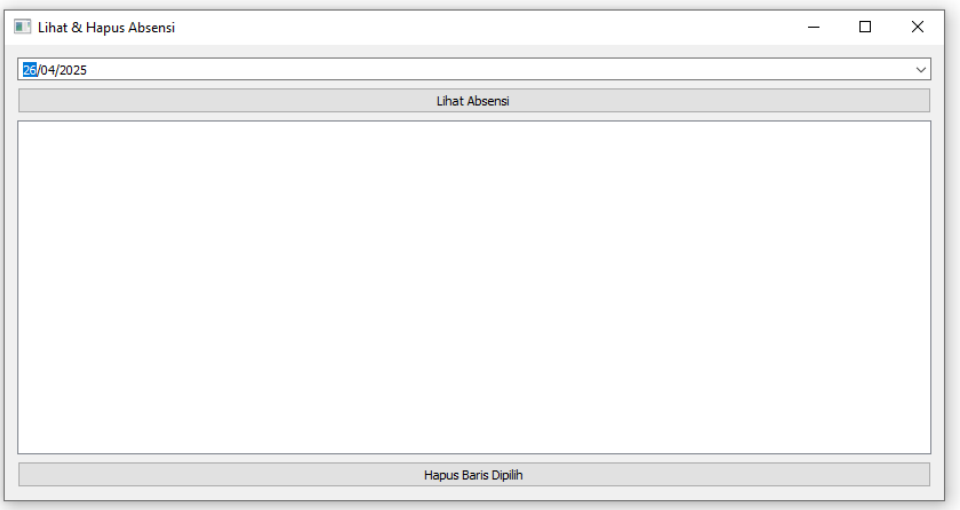
```
            self.ax.plot(history.history['val_accuracy'],
label='Val Accuracy')
        if 'loss' in history.history:
            self.ax.plot(history.history['loss'], label='Train
Loss')
        if 'val_loss' in history.history:
            self.ax.plot(history.history['val_loss'], label='Val
Loss')

        self.ax.set_title("Training Accuracy & Loss")
        self.ax.set_xlabel("Epoch")
        self.ax.set_ylabel("Value")
        self.ax.legend()
        self.canvas.draw()
```

Kode python ini merupakan Script ini adalah jendela absensi wajah berbasis PyQt5 yang menggunakan YOLOv8 untuk deteksi wajah dan DeepFace untuk verifikasi, lalu mencatat data kehadiran siswa ke file Excel beserta foto dan informasi guru serta mata pelajaran.

Menu yang ditampilkan pada gambar merupakan tampilan dari fitur **Absensi Otomatis** dalam sistem absensi berbasis face recognition. Pada antarmuka ini, sistem secara otomatis mengecek jadwal yang sedang berlangsung berdasarkan data yang telah diinput sebelumnya. Jika tidak ditemukan jadwal yang aktif sesuai waktu saat ini, maka akan muncul pesan notifikasi berupa ikon peringatan dan teks bertuliskan **"Tidak ada jadwal aktif saat ini."**. Tampilan ini menunjukkan bahwa tidak ada sesi pelajaran yang sedang berlangsung, sehingga proses absensi otomatis belum bisa dilakukan. Fungsi utama dari menu ini adalah memastikan bahwa proses absensi hanya dilakukan pada waktu yang sesuai dengan jadwal yang telah ditentukan sebelumnya, sehingga sistem berjalan efisien dan sesuai konteks.

7. Lihat Data Absensi



Gambar 4.16 Lihat Data Absensi

```
import os
import pandas as pd
from PyQt5.QtWidgets import (
    QWidget, QVBoxLayout, QTableWidget, QTableWidgetItem,
    QPushButton, QDateEdit, QMessageBox, QHBoxLayout, QLabel
)
from PyQt5.QtGui import QPixmap
from PyQt5.QtCore import QDate, Qt
from datetime import datetime
class LihatAbsensiWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Lihat & Hapus Absensi")
        self.setGeometry(200, 200, 1000, 500)
        layout = QVBoxLayout()
        # Pilih tanggal
        self.date_edit = QDateEdit()
        self.date_edit.setCalendarPopup(True)
        self.date_edit.setDate(QDate.currentDate())
        layout.addWidget(self.date_edit)
        # Tombol lihat
        self.load_btn = QPushButton("Lihat Absensi")
        self.load_btn.clicked.connect(self.load_data)
        layout.addWidget(self.load_btn)
        # Tabel absensi
        self.table = QTableWidget()
        layout.addWidget(self.table)
        # Tombol hapus baris
        btn_layout = QHBoxLayout()
        self.delete_btn = QPushButton("Hapus Baris Dipilih")
        self.delete_btn.clicked.connect(self.hapus_baris)
        btn_layout.addWidget(self.delete_btn)
        layout.addLayout(btn_layout)
        self.setLayout(layout)
    def load_data(self):
        tanggal = self.date_edit.date().toString("yyyy-MM-
dd")
        file_path = f"data/absensi/{tanggal}.xlsx"
```

```
        if not os.path.exists(file_path):
            QMessageBox.warning(self, "Data Tidak Ditemukan",
f"Tidak ada data untuk {tanggal}")
            return
        self.df = pd.read_excel(file_path)
        self.table.setRowCount(len(self.df))
        self.table.setColumnCount(len(self.df.columns))
        self.table.setHorizontalHeaderLabels(self.df.columns)
        for i in range(len(self.df)):
            for j in range(len(self.df.columns)):
                column_name = self.df.columns[j]
                if column_name == "Foto":
                    foto_path = str(self.df.iat[i, j])
                    if os.path.exists(foto_path):
                        pixmap =
QPixmap(foto_path).scaled(100, 100, Qt.KeepAspectRatio)
                        label = QLabel()
                        label.setPixmap(pixmap)
                        label.setAlignment(Qt.AlignCenter)
                        self.table.setCellWidget(i, j, label)
                    else:
                        self.table.setItem(i, j,
QTableWidgetItem("Foto tidak ditemukan"))
                else:
                    self.table.setItem(i, j,
QTableWidgetItem(str(self.df.iat[i, j])))

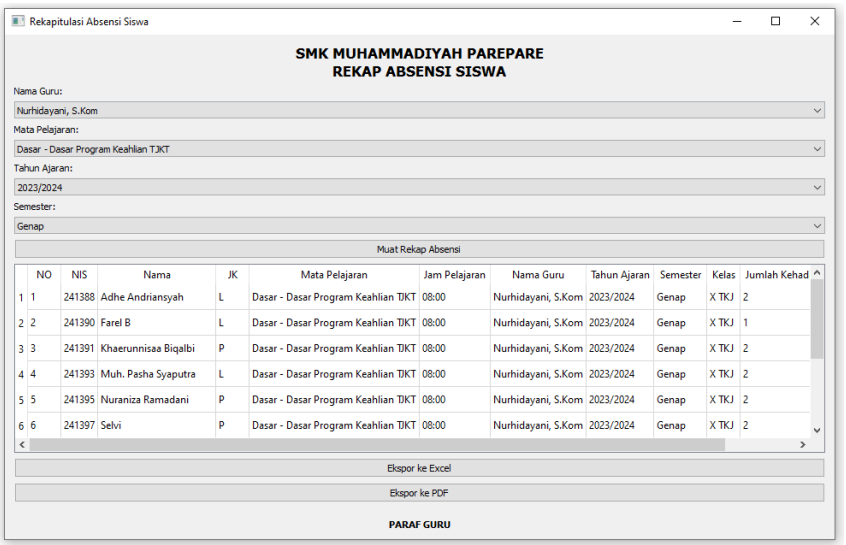
        self.table.resizeColumnsToContents()
        self.table.resizeRowsToContents()
    def hapus_baris(self):
        selected = self.table.currentRow()
        if selected == -1:
            QMessageBox.information(self, "Info", "Pilih
baris yang ingin dihapus")
            return
        confirm = QMessageBox.question(self, "Konfirmasi",
"Yakin ingin menghapus baris ini?")
        if confirm != QMessageBox.Yes:
            return
        self.df.drop(index=selected, inplace=True)
        tanggal = self.date_edit.date().toString("yyyy-MM-
dd")
        file_path = f"data/absensi/{tanggal}.xlsx"
        self.df.reset_index(drop=True, inplace=True)
        self.df.to_excel(file_path, index=False)
        self.load_data()
```

Script ini adalah antarmuka PyQt5 untuk menampilkan dan mengelola data absensi berdasarkan tanggal tertentu dari file Excel, termasuk menampilkan foto wajah siswa dan memungkinkan penghapusan baris absensi yang dipilih.

Menu yang ditampilkan pada gambar merupakan tampilan dari fitur Lihat & Hapus Absensi dalam sistem absensi berbasis face recognition. Pada bagian atas, terdapat komponen pemilih tanggal yang memungkinkan pengguna untuk memilih

tanggal tertentu guna menampilkan data absensi yang terjadi pada hari tersebut. Setelah memilih tanggal, pengguna dapat mengklik tombol "Lihat Absensi" untuk menampilkan daftar absensi yang tersimpan, yang nantinya akan muncul dalam area tabel di bagian tengah jendela. Jika terdapat data yang tidak valid atau perlu dikoreksi, pengguna dapat memilih baris tertentu dan menekan tombol "Hapus Baris Dipilih" di bagian bawah untuk menghapus entri tersebut. Menu ini memberikan kemudahan bagi admin atau operator untuk melakukan pengecekan dan pengelolaan data absensi secara manual, termasuk penghapusan data yang tidak relevan atau duplikat, sehingga kualitas dan keakuratan data tetap terjaga.

8. Rekapitulasi Absensi



Gambar 4.17 Rekapitulasi Absensi

```
import os
import pandas as pd
from PyQt5.QtWidgets import (
          QWidget,    QVBoxLayout,    QLabel,    QTableWidget,
QTableWidgetItem,
        QPushButton,   QFileDialog,   QMessageBox,   QComboBox,
QApplication
)
from PyQt5.QtCore import Qt
from reportlab.lib.pagesizes import A4
from reportlab.lib import colors
from  reportlab.platypus  import  SimpleDocTemplate,  Table,
TableStyle, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet
import sys

class RekapWindow(QWidget):
```

```
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Rekapitulasi Absensi Siswa")
        self.setGeometry(200, 200, 1000, 600)

        layout = QVBoxLayout()

        # Header Sekolah
        header = QLabel("<h2>SMK MUHAMMADIYAH PAREPARE<br>REKAP
ABSENSI SISWA</h2>")
        header.setAlignment(Qt.AlignCenter)
        layout.addWidget(header)

        # Input ComboBox Guru, Mapel, Tahun Ajaran, Semester
        self.combo_guru = QComboBox()
        self.combo_mapel = QComboBox()
        self.combo_tahun_ajaran = QComboBox()
        self.combo_semester = QComboBox()

        # Tambahkan pilihan tahun ajaran
                self.combo_tahun_ajaran.addItems(["2023/2024",
"2024/2025", "2025/2026"])
        # Tambahkan pilihan semester
        self.combo_semester.addItems(["Ganjil", "Genap"])

        layout.addWidget(QLabel("Nama Guru:"))
        layout.addWidget(self.combo_guru)
        layout.addWidget(QLabel("Mata Pelajaran:"))
        layout.addWidget(self.combo_mapel)
        layout.addWidget(QLabel("Tahun Ajaran:"))
        layout.addWidget(self.combo_tahun_ajaran)
        layout.addWidget(QLabel("Semester:"))
        layout.addWidget(self.combo_semester)

        # Muat jadwal Excel
        self.load_jadwal()

        # Tombol Load Rekap
        self.load_btn = QPushButton("Muat Rekap Absensi")
        self.load_btn.clicked.connect(self.muat_rekap)
        layout.addWidget(self.load_btn)

        # Tabel Absensi
        self.table = QTableWidget()
        layout.addWidget(self.table)

        # Tombol Ekspor Excel
        self.export_btn = QPushButton("Ekspor ke Excel")
        self.export_btn.clicked.connect(self.ekspor_data)
        layout.addWidget(self.export_btn)

        # Tombol Ekspor PDF
        self.export_pdf_btn = QPushButton("Ekspor ke PDF")
        self.export_pdf_btn.clicked.connect(self.ekspor_pdf)
        layout.addWidget(self.export_pdf_btn)

        # Footer (Paraf Guru)
        footer = QLabel("\nPARAF GURU")
        footer.setStyleSheet("font-weight: bold;")
        footer.setAlignment(Qt.AlignCenter)
        layout.addWidget(footer)
```

```
        self.setLayout(layout)

    def load_jadwal(self):
        """Load daftar guru & mapel dari file jadwal Excel"""
        jadwal_file = "data/jadwal.xlsx"
        if not os.path.exists(jadwal_file):
           QMessageBox.warning(self, "Jadwal Tidak Ditemukan",
f"File {jadwal_file} tidak ada.")
            return

        try:
            df_jadwal = pd.read_excel(jadwal_file)
            required_cols = {"Nama Guru", "Mata Pelajaran"}
            if not required_cols.issubset(df_jadwal.columns):
                   QMessageBox.critical(self, "Error", f"File
jadwal harus punya kolom: {', '.join(required_cols)}")
                return

               self.combo_guru.addItems(sorted(df_jadwal["Nama
Guru"].dropna().unique().tolist()))
              self.combo_mapel.addItems(sorted(df_jadwal["Mata
Pelajaran"].dropna().unique().tolist()))

        except Exception as e:
           QMessageBox.critical(self, "Error", f"Gagal membaca
jadwal: {e}")

    def muat_rekap(self):
        folder = "data/absensi"
        all_data = []

        for filename in os.listdir(folder):
            if filename.endswith(".xlsx"):
                filepath = os.path.join(folder, filename)
                try:
                    df = pd.read_excel(filepath)
                    all_data.append(df)
                except Exception as e:
                    print(f"Gagal membaca {filename}: {e}")
                    continue

        if not all_data:
           QMessageBox.warning(self, "Kosong", "Tidak ada data
absensi ditemukan.")
            return

        df_all = pd.concat(all_data, ignore_index=True)

         required_cols = {'Nama', 'NIS', 'JK', 'Kelas', 'Mata
Pelajaran', 'Jam Pelajaran', 'Tanggal', 'Status', 'Nama Guru'}
        if not required_cols.issubset(df_all.columns):
            QMessageBox.critical(self, "Error", f"File absensi
harus memiliki kolom: {', '.join(required_cols)}")
            return

        # Filter sesuai guru, mapel, tahun ajaran, dan semester
        guru = self.combo_guru.currentText()
        mapel = self.combo_mapel.currentText()
        tahun_ajaran = self.combo_tahun_ajaran.currentText()
        semester = self.combo_semester.currentText()
```

```
        df_filtered = df_all[(df_all['Nama Guru'] == guru) &
(df_all['Mata Pelajaran'] == mapel)]

        if df_filtered.empty:
            QMessageBox.information(self, "Info", "Tidak ada
data absensi untuk guru/mapel yang dipilih.")
            self.table.clear()
            return

        # Filter bulan sesuai semester
                                    df_filtered['Bulan']       =
pd.to_datetime(df_filtered['Tanggal']).dt.month
        if semester == "Ganjil":
                                                df_filtered      =
df_filtered[df_filtered['Bulan'].between(7, 12)]
        else:  # Genap
                                                df_filtered      =
df_filtered[df_filtered['Bulan'].between(1, 6)]

        if df_filtered.empty:
            QMessageBox.information(self, "Info", "Tidak ada
data absensi di semester yang dipilih.")
            self.table.clear()
            return

        # Hitung jumlah hadir
        hadir_count = (
            df_filtered[df_filtered['Status'] == "Hadir"]
            .groupby(['NIS', 'Nama', 'JK', 'Kelas']).size()
            .reset_index(name="Jumlah Kehadiran")
        )

          jam = df_filtered['Jam Pelajaran'].iloc[0] if not
df_filtered.empty else ""

        df_rekap = hadir_count.copy()
        df_rekap.insert(0, "NO", range(1, len(df_rekap) + 1))
        df_rekap.insert(4, "Mata Pelajaran", mapel)
        df_rekap.insert(5, "Jam Pelajaran", jam)
        df_rekap.insert(6, "Nama Guru", guru)
        df_rekap.insert(7, "Tahun Ajaran", tahun_ajaran)
        df_rekap.insert(8, "Semester", semester)

        self.df_rekap = df_rekap
        self.df_all = df_filtered  # simpan untuk PDF

        # Tampilkan di tabel PyQt
        self.table.setRowCount(len(df_rekap))
        self.table.setColumnCount(len(df_rekap.columns))
        self.table.setHorizontalHeaderLabels(df_rekap.columns)

        for i in range(len(df_rekap)):
            for j in range(len(df_rekap.columns)):
                                        self.table.setItem(i,   j,
QTableWidgetItem(str(df_rekap.iat[i, j])))

        self.table.resizeColumnsToContents()

    def ekspor_data(self):
        if not hasattr(self, 'df_rekap'):
```

```
            QMessageBox.information(self, "Info", "Data belum
dimuat.")
            return

         path, _ = QFileDialog.getSaveFileName(self, "Simpan
File", "rekap_absensi.xlsx", "Excel Files (*.xlsx)")
        if path:
            self.df_rekap.to_excel(path, index=False)
               QMessageBox.information(self, "Sukses", f"Data
berhasil diekspor ke:\n{path}")

    def ekspor_pdf(self):
        if not hasattr(self, 'df_rekap'):
             QMessageBox.information(self, "Info", "Data belum
dimuat.")
            return

         path, _ = QFileDialog.getSaveFileName(self, "Simpan
PDF", "rekap_absensi.pdf", "PDF Files (*.pdf)")
        if not path:
            return

        doc = SimpleDocTemplate(path, pagesize=A4)
        styles = getSampleStyleSheet()
        elements = []

        # Header
             elements.append(Paragraph("<b>ABSENSI SISWA SMK
MUHAMMADIYAH PAREPARE</b>", styles['Title']))
        elements.append(Spacer(1, 12))

        # Info tambahan
                elements.append(Paragraph(f"Tahun Ajaran :
{self.combo_tahun_ajaran.currentText()}", styles['Normal']))
                    elements.append(Paragraph(f"Semester :
{self.combo_semester.currentText()}", styles['Normal']))
                elements.append(Paragraph(f"Nama Guru :
{self.combo_guru.currentText()}", styles['Normal']))
              elements.append(Paragraph(f"Mata Pelajaran :
{self.combo_mapel.currentText()}", styles['Normal']))
               elements.append(Paragraph(f"Jam Pelajaran :
{self.df_rekap['Jam Pelajaran'].iloc[0]}", styles['Normal']))
                       elements.append(Paragraph(f"Kelas :
{self.df_rekap['Kelas'].iloc[0]}", styles['Normal']))
        elements.append(Spacer(1, 12))

        # Ambil semua tanggal absensi unik
                                        tanggal_list =
sorted(self.df_all['Tanggal'].unique().tolist())

        # Buat header tabel
         table_header = ["NO", "NIS", "NAMA SISWA", "JK"] +
tanggal_list + ["Jumlah Kehadiran"]
        table_data = [table_header]

        # Isi data siswa
        for i, row in self.df_rekap.iterrows():
             nis, nama, jk, kelas = row["NIS"], row["Nama"],
row["JK"], row["Kelas"]
            hadir_per_tanggal = []
            for tgl in tanggal_list:
```

```
                subset = self.df_all[(self.df_all['Tanggal'] ==
tgl) & (self.df_all['NIS'] == nis)]
                  if not subset.empty and (subset['Status'] ==
"Hadir").any():
                     hadir_per_tanggal.append("✓")
                 else:
                     hadir_per_tanggal.append("✗")
            jumlah = row["Jumlah Kehadiran"]
                  table_data.append([i+1,  nis,  nama,  jk]  +
hadir_per_tanggal + [jumlah])

        # Tambahkan baris Paraf Guru
         paraf_row = ["", "", "PARAF GURU", ""]  # kolom awal
(NO, NIS, Nama, JK)
        paraf_tanggal = []
        for tgl in tanggal_list:
              subset  =  self.df_all[(self.df_all['Tanggal']  ==
tgl)        &        (self.df_all['Nama        Guru']        ==
self.combo_guru.currentText())]
            if not subset.empty:
                paraf_tanggal.append("✓")
            else:
                paraf_tanggal.append("")
         paraf_row += paraf_tanggal + [""]  # kolom terakhir
jumlah kehadiran kosong
        table_data.append(paraf_row)

        # Buat tabel PDF
        table = Table(table_data, repeatRows=1)
        table.setStyle(TableStyle([
            ('BACKGROUND', (0,0), (-1,0), colors.lightgrey),
            ('GRID', (0,0), (-1,-1), 1, colors.black),
            ('ALIGN', (0,0), (-1,-1), 'CENTER'),
            ('VALIGN', (0,0), (-1,-1), 'MIDDLE'),
        ]))

        elements.append(table)
        doc.build(elements)

        QMessageBox.information(self, "Sukses", f"PDF berhasil
disimpan di:\n{path}")

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = RekapWindow()
    window.show()
    sys.exit(app.exec_())
```

Script ini adalah program GUI Rekapitulasi Absensi Siswa yang dibuat dengan PyQt5, menggunakan Pandas untuk membaca data Excel, dan ReportLab untuk mengekspor laporan dalam bentuk PDF.

Menu dari sebuah aplikasi Rekapitulasi Absensi Siswa yang digunakan di SMK Muhammadiyah Parepare. Pada bagian atas terdapat judul aplikasi yang menunjukkan

fungsi utama yaitu untuk merekap data kehadiran siswa. Aplikasi ini memiliki beberapa pilihan filter berupa nama guru, mata pelajaran, tahun ajaran, dan semester. Setelah filter dipilih, pengguna dapat menekan tombol "Muat Rekap Absensi" untuk menampilkan data kehadiran siswa.

Data absensi siswa ditampilkan dalam bentuk tabel yang berisi informasi nomor urut, NIS, nama siswa, jenis kelamin, mata pelajaran, jam pelajaran, nama guru, tahun ajaran, semester, kelas, serta jumlah kehadiran siswa. Selain itu, aplikasi juga menyediakan fitur ekspor ke Excel dan ekspor ke PDF, sehingga laporan absensi dapat disimpan atau dicetak. Di bagian bawah terdapat opsi paraf guru yang berfungsi untuk memberikan tanda tangan digital guru pada laporan absensi, terutama ketika diekspor dalam bentuk PDF.

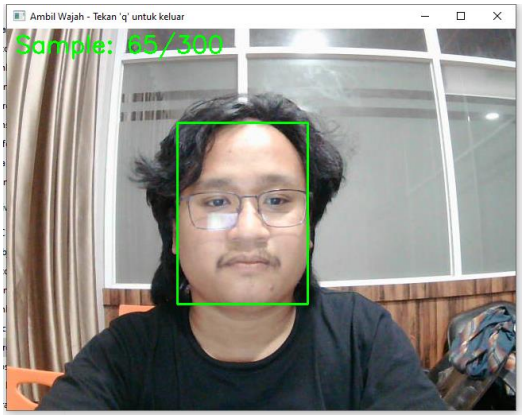
Dengan demikian, aplikasi ini membantu guru dalam mengelola, merekap, dan mencetak data kehadiran siswa secara lebih mudah dan rapi.

D. Arsitektur Model

Absensi ini menggunakan dua sistem arsitektur yaitu YOLOV8 dan Deepface. YOLOV8 digunakan untuk deteksi wajah dengan penggunaan bounding box menentukan lokasi wajah dan memungkinkan memotong gambar wajah, bounding box jg digunakan untuk memberikan tanda visual pada GUI agar meningkatkan efisiensi dan akurasi proses. Sedangkan Deepface digunakan untuk mengubah wajah menjadi embedding yang bisa dibandingkan satu sama lain.

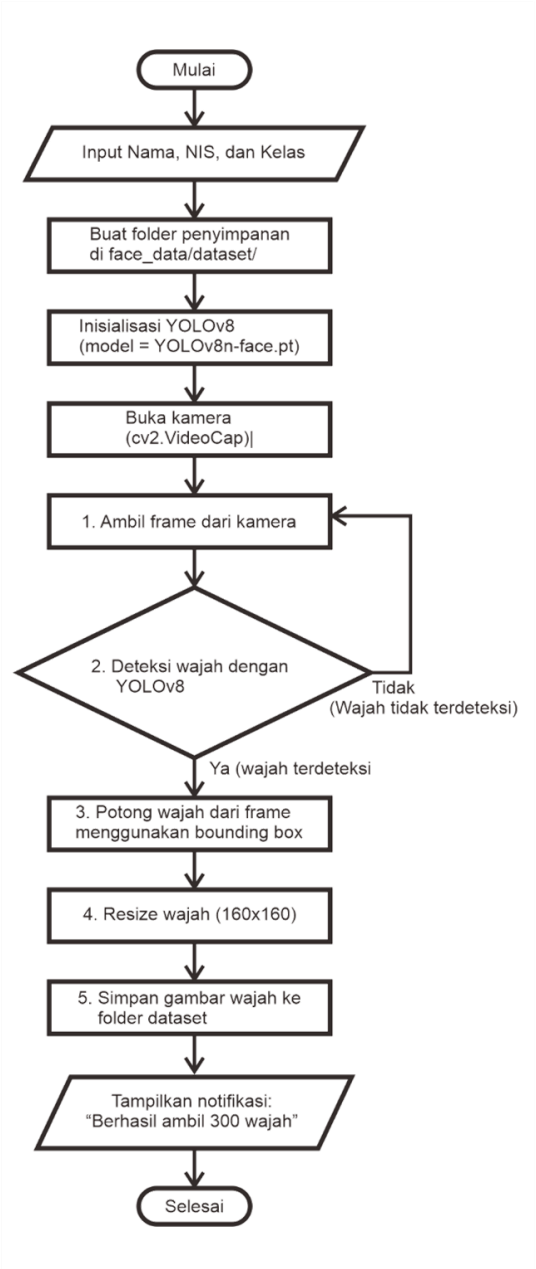
1. YOLOV8

YOLOv8 (You Only Look Once versi 8) adalah model deteksi objek berbasis deep learning yang cepat dan akurat.



Gambar 4.18 Pengambilan Wajah Menggunakan YOLOV8

Gambar menampilkan Proses pengambilan gambar wajah untuk sistem absensi menggunakan YOLOV8 secara otomatis.



Gambar 4.19 Flowchart cara kerja yolov8

Proses dimulai ketika pengguna memasukkan data siswa, yang terdiri dari nama, NIS (Nomor Induk Siswa), dan kelas. Setelah data dimasukkan, sistem akan membuat folder penyimpanan baru yang terletak di direktori face_data/dataset/, dan folder ini akan digunakan untuk menyimpan hasil gambar wajah.

Selanjutnya, sistem menginisialisasi model deteksi wajah menggunakan YOLOv8, tepatnya model YOLOv8n-face.pt. Setelah model berhasil dimuat, kamera akan diaktifkan menggunakan cv2.VideoCapture untuk mengambil video secara langsung.

Sistem kemudian akan secara terus-menerus mengambil frame dari kamera. Setiap frame yang diambil akan dianalisis menggunakan model YOLOv8 untuk mendeteksi keberadaan wajah. Jika pada suatu frame tidak ditemukan wajah sesuai data wajah rujukan dalam database, maka sistem akan kembali mengambil frame berikutnya.

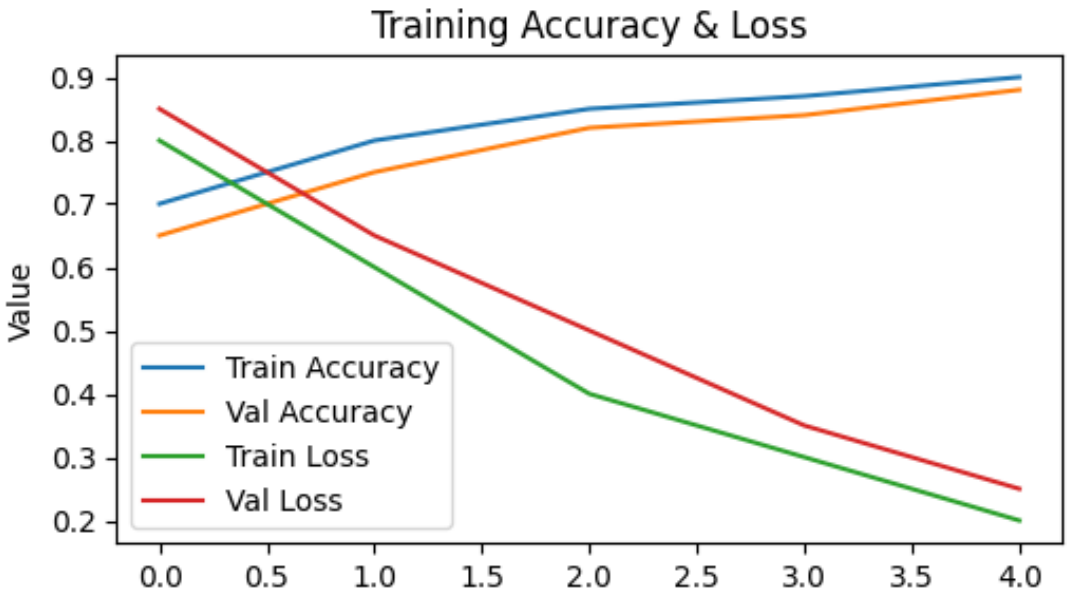
Namun, jika YOLOv8 berhasil mendeteksi wajah pada sebuah frame, sistem akan memotong bagian wajah dari frame tersebut menggunakan informasi bounding box yang diberikan oleh model. Wajah yang telah dipotong ini kemudian diubah ukurannya menjadi 160x160 piksel agar sesuai dengan standar ukuran dataset.

Setelah itu, gambar wajah yang telah diubah ukurannya akan disimpan ke dalam folder dataset yang sebelumnya telah dibuat. Proses pengambilan dan penyimpanan ini akan terus berlangsung hingga sistem berhasil menyimpan sebanyak 300 gambar wajah.

Setelah jumlah gambar yang diambil mencapai 300, sistem akan menampilkan notifikasi yang menyatakan: "Berhasil ambil 300 wajah", dan proses pun selesai.

2. Deepface

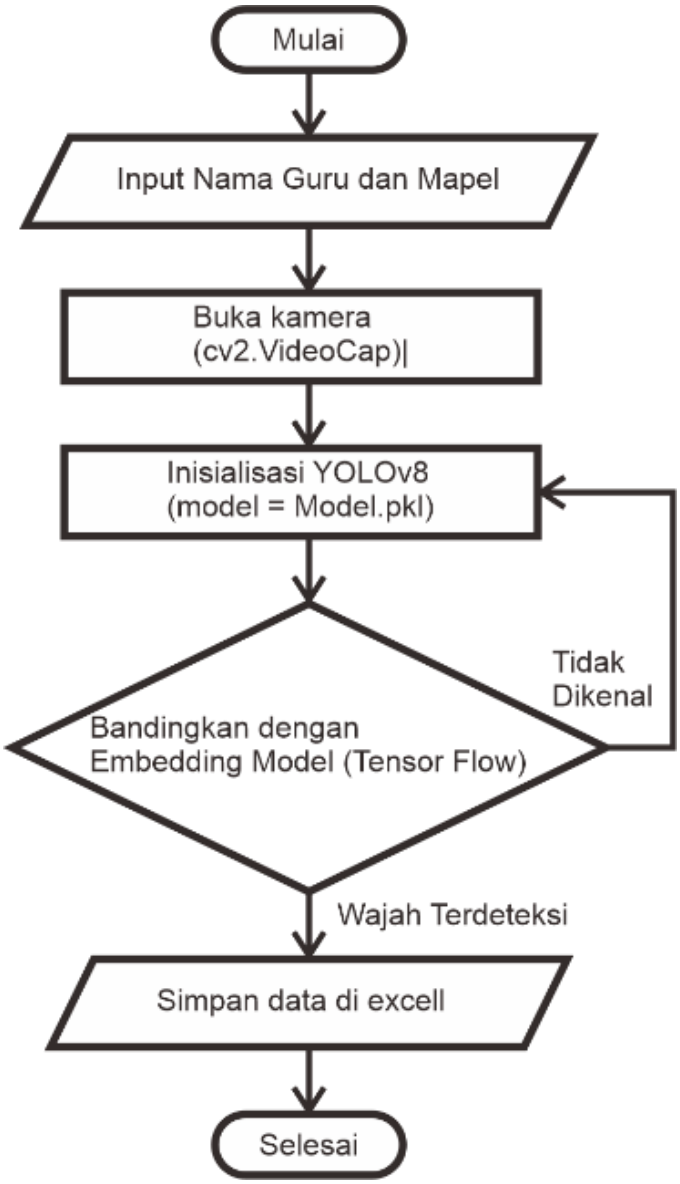
DeepFace mengubah gambar wajah menjadi embedding (vektor numerik 128D/512D, tergantung model).



Gambar 4.20 Hasil Training Menggunakan Deepface

Gambar menampilkan hasil training menggunakan deepface untuk sistem mengubah gambar wahag menjadi embedding.

3. Tensorflow



Gambar 4.21 Flowchar Cara kerja tensorflow

Proses diawali dengan menginputkan nama guru dan mata pelajaran yang sedang berlangsung. Informasi ini nantinya akan disimpan bersama data absensi. Setelah itu, kamera diaktifkan menggunakan cv2.VideoCapture untuk mulai menangkap wajah siswa secara real-time. Selanjutnya, sistem akan menginisialisasi model YOLOv8 yang sudah dilatih sebelumnya untuk mendeteksi wajah (misalnya model model.pkl).

Begitu kamera aktif dan model YOLOv8 siap, sistem akan mendeteksi wajah dalam frame yang diambil dari kamera, kemudian hasil deteksi ini akan dibandingkan dengan

data wajah yang sudah disimpan dalam bentuk embedding menggunakan model yang telah dilatih dengan TensorFlow.

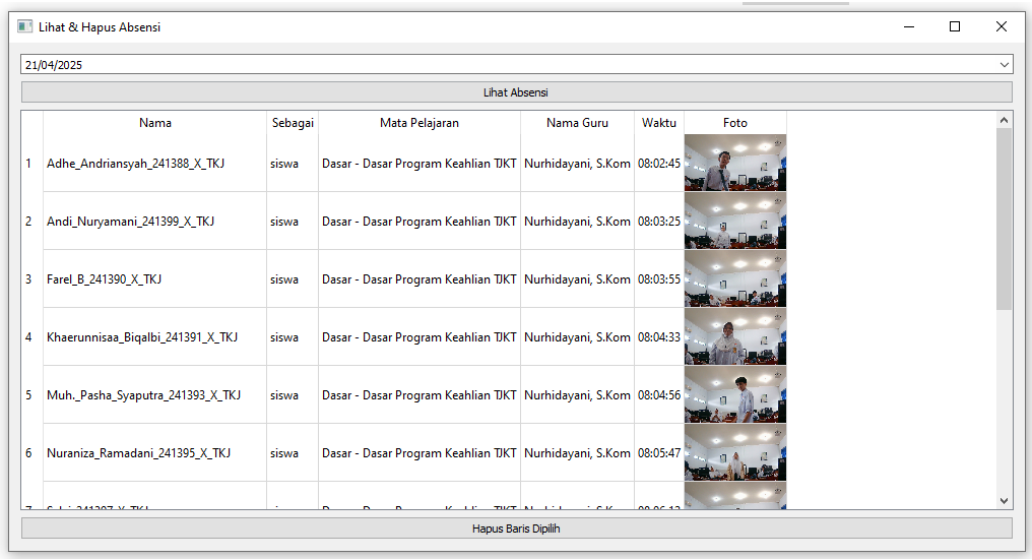
Jika wajah yang terdeteksi **tidak cocok atau tidak ditemukan dalam database** (dengan kata lain tidak dikenal), maka sistem akan mengabaikannya dan kembali menunggu wajah lain.

Namun, jika wajah yang terdeteksi **cocok dengan salah satu data embedding**, maka wajah tersebut dianggap dikenali, dan sistem akan **menyimpan data absensi ke dalam file Excel**. Data yang disimpan meliputi informasi siswa, waktu kehadiran, nama guru, dan nama mata pelajaran. Setelah data berhasil disimpan, proses absensi dianggap selesai.

E. Hasil Uji Coba Aplikasi

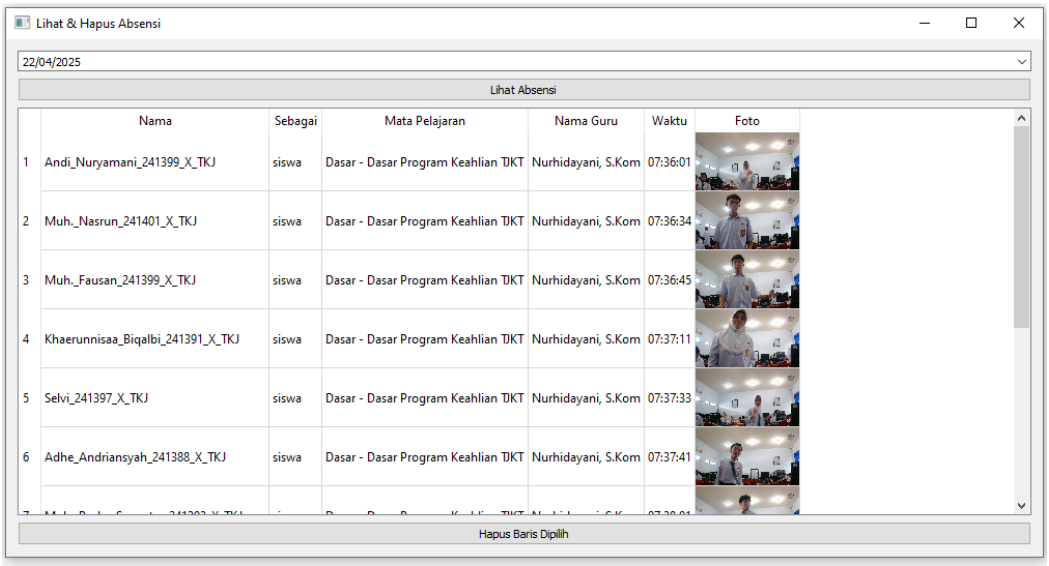
Berikut ini adalah hasil uji coba aplikasi pada sekolah SMKS Muhammadiyah Kota Parepare. Uji coba dilakukan untuk menguji keakuratan dan keandalan sistem absensi berbasis face recognition menggunakan kombinasi YOLOv8 sebagai detektor wajah dan DeepFace (Facenet) sebagai pengenal identitas wajah. Pengujian dilakukan pada laptop Windows 10 dengan Python 3.10, menggunakan kamera eksternal dan 14 wajah terdaftar. Wajah diuji Selama 5 Hari dalam waktu berbeda menyesuaikan dengan jadwal sekolah.

1. Hari Pertama



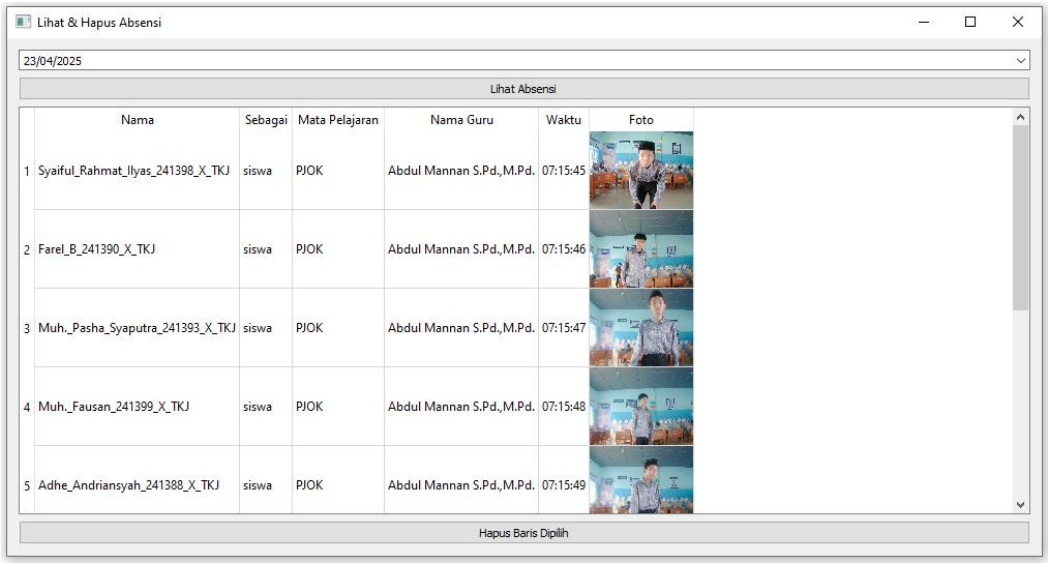
Gambar 4.22 Absensi 21-04-2025

2. Hari Kedua



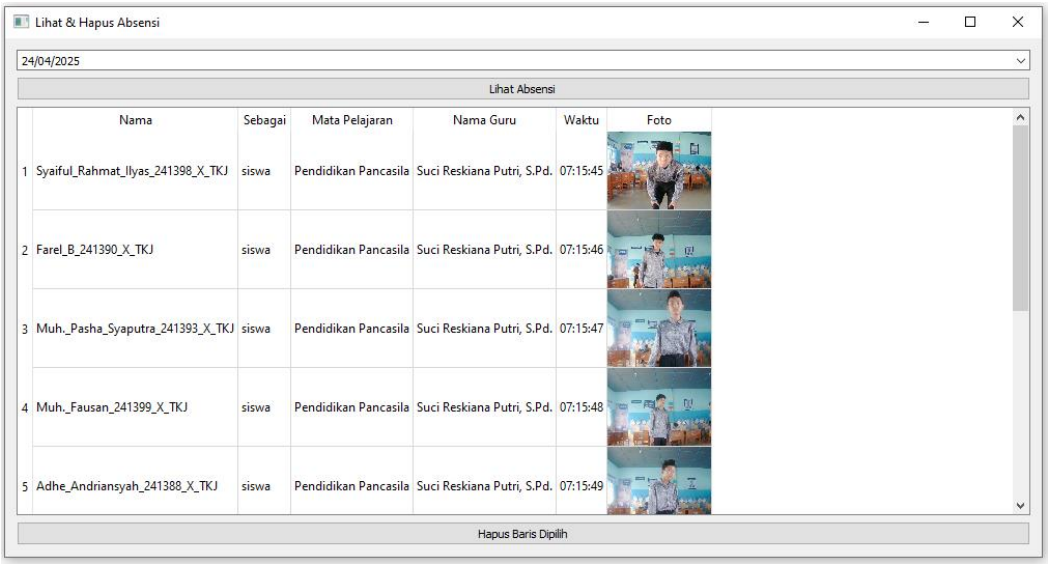
Gambar 4.23 Absensi 22-04-2025

3. Hari Ketiga



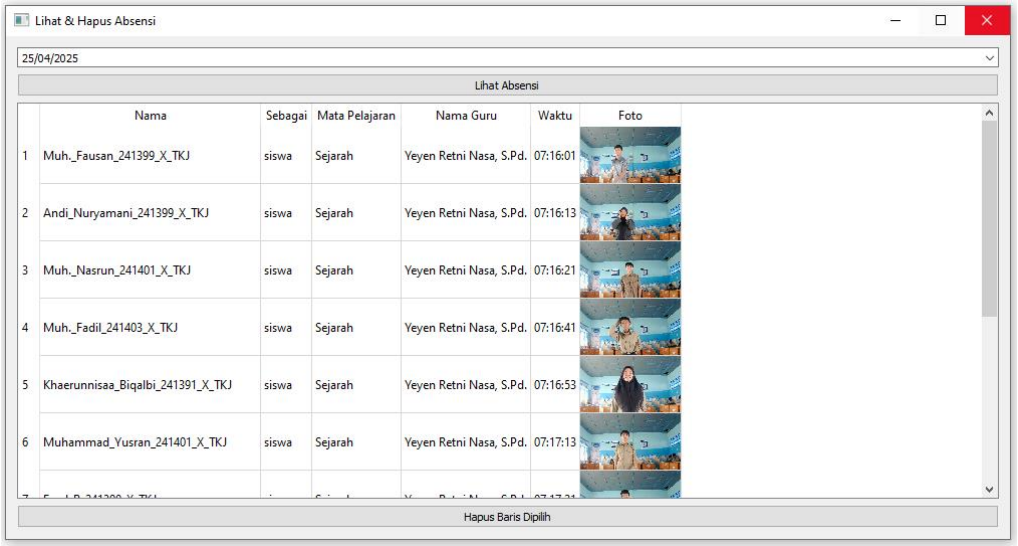
Gambar 4.24 Absensi 23-04-2025

4. Hari Keempat



Gambar 4.25 Absensi 24-04-2025

5. Hari Kelima



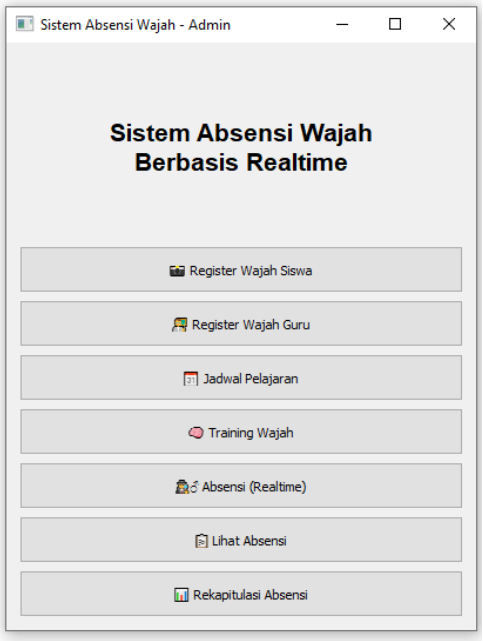
Gambar 4.26 Absensi 25-04-2025

F. PENGUJIAN SISTEM

1. Pengujian Blackbox

a. Halaman Utama

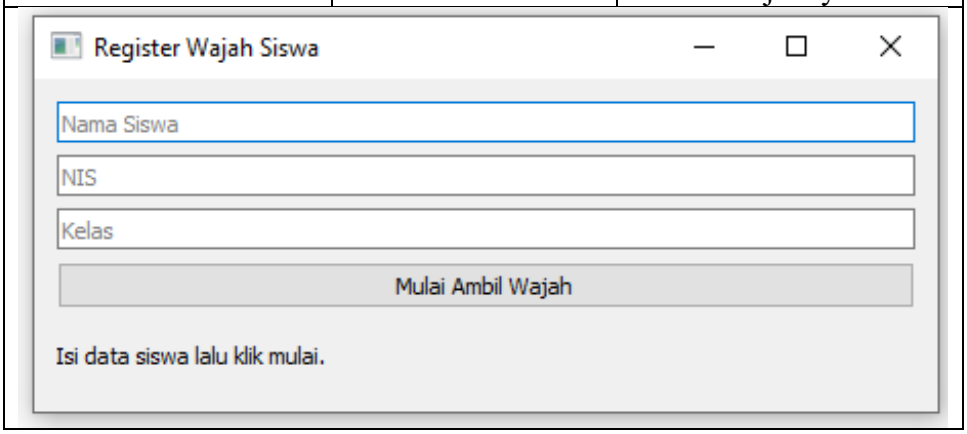
Test Faktor	Hasil	Keterangan
Menu Utama	✓	Berhasil, aplikasi dapat menampilkan halaman utama



b. Form Registrasi Wajah Siswa

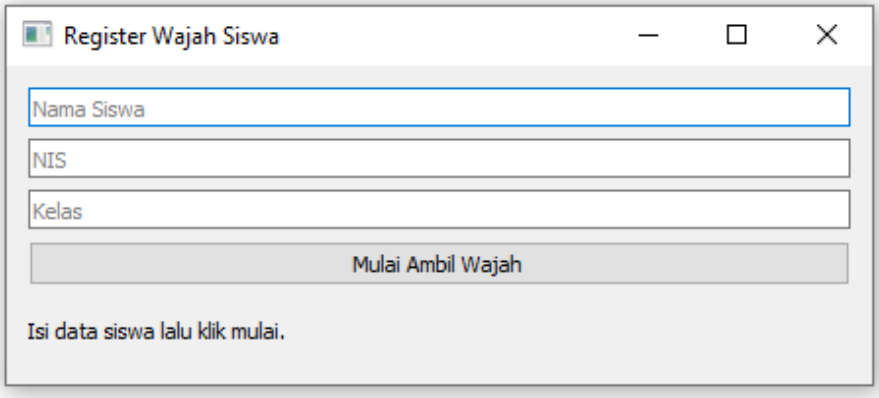
Test Faktor	Hasil	Keterangan

Registrasi Wajah Siswa	✔	Berhasil, aplikasi dapat menginpunt data siswa yang akan diambil data wajahnya



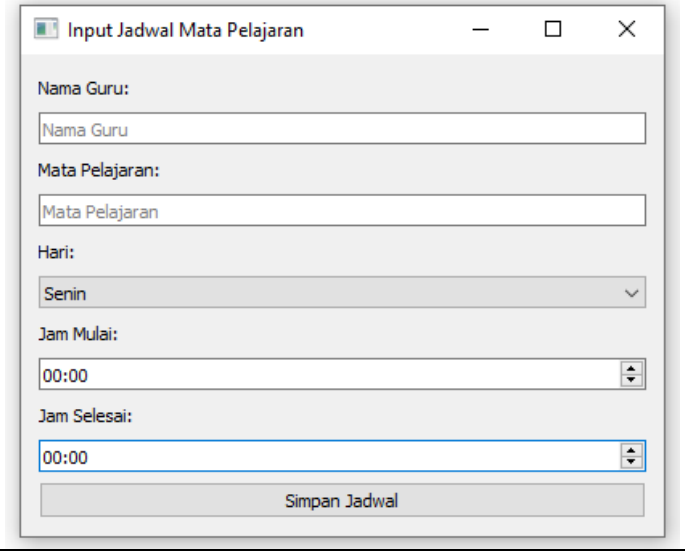
c. Form Registrasi Wajah Guru

Test Faktor	Hasil	Keterangan
Registrasi Wajah Guru	✔	Berhasil, aplikasi dapat menginpunt data guru yang akan diambil data wajahnya

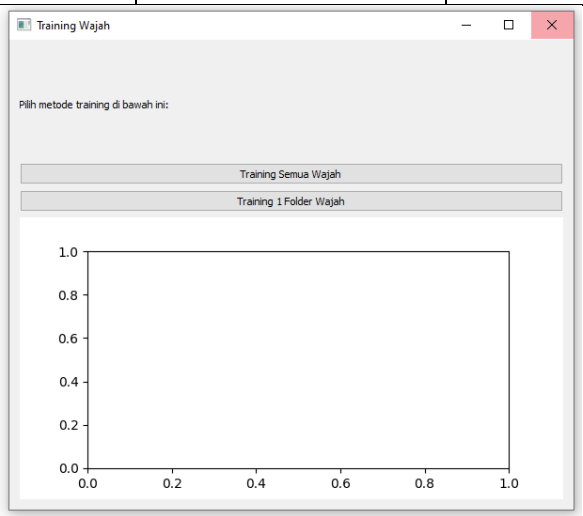


d. Fitur Jadwal Pelajaran

Test Faktor	Hasil	Keterangan
Input Jadwal Pelajaran	✔	Berhasil, aplikasi dapat menginpunt

		data Jadwal Pelajaran beserta jadwalnya
		

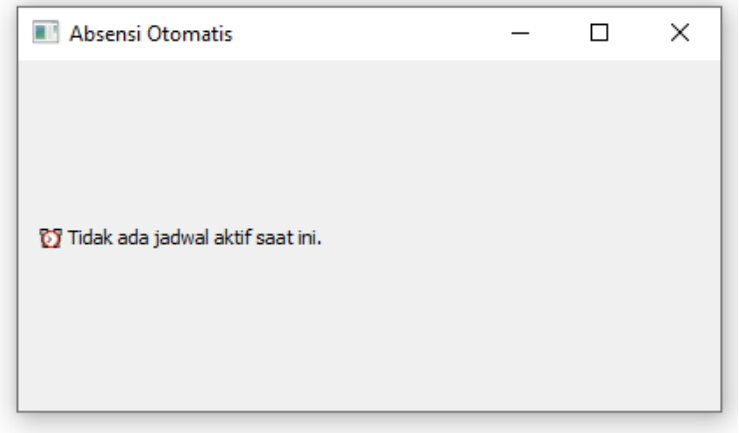
e. Fitur Training Wajah

Test Faktor	Hasil	Keterangan
Training Wajah	✔	Berhasil, training wajah berhasil dilakukan dan menyimpan ke model.pkl
		

f. Rekam Absensi

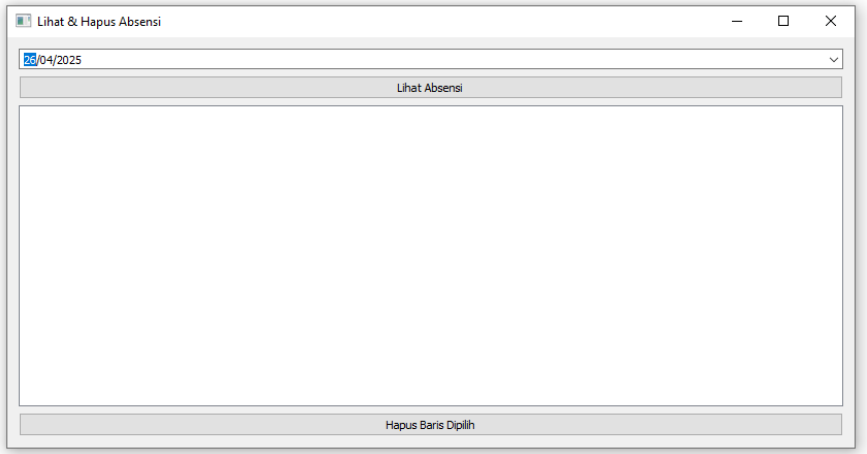
Test Faktor	Hasil	Keterangan
Rekam Absensi	✔	Berhasil, Kamera aktif dan absensi dilakukan otomatis.pelajaran yang sebelumnya telah di input berhasil digunakan sesuai jadwalnya. absensi dan dapat

		mengambil wajah siswa yang masuk dalam kamera



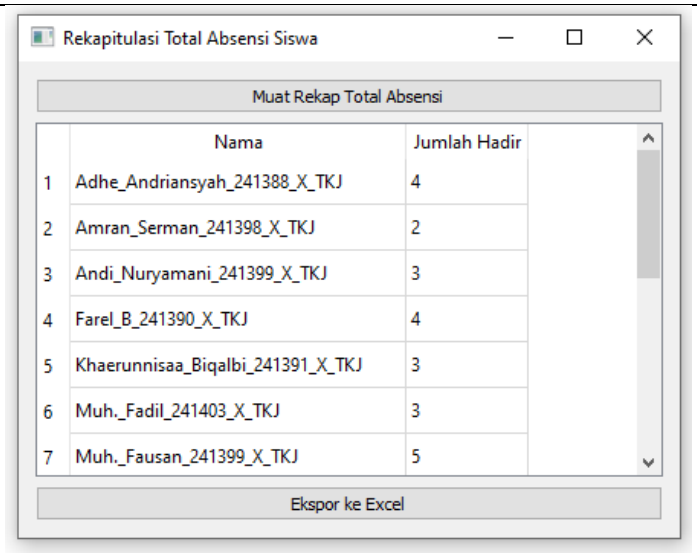
g. Lihat Data Absensi

Test Faktor	Hasil	Keterangan
Rekam Absensi	✓	Berhasil, Tabel absensi tampil sesuai tanggal dengan output yang sesuai



h. Rekapitulasi Absensi

Test Faktor	Hasil	Keterangan
Rekap Absensi	✓	Berhasil, Data siswa dan jumlah kehadiran tampil dan File Excel berhasil dibuat dan disimpan.



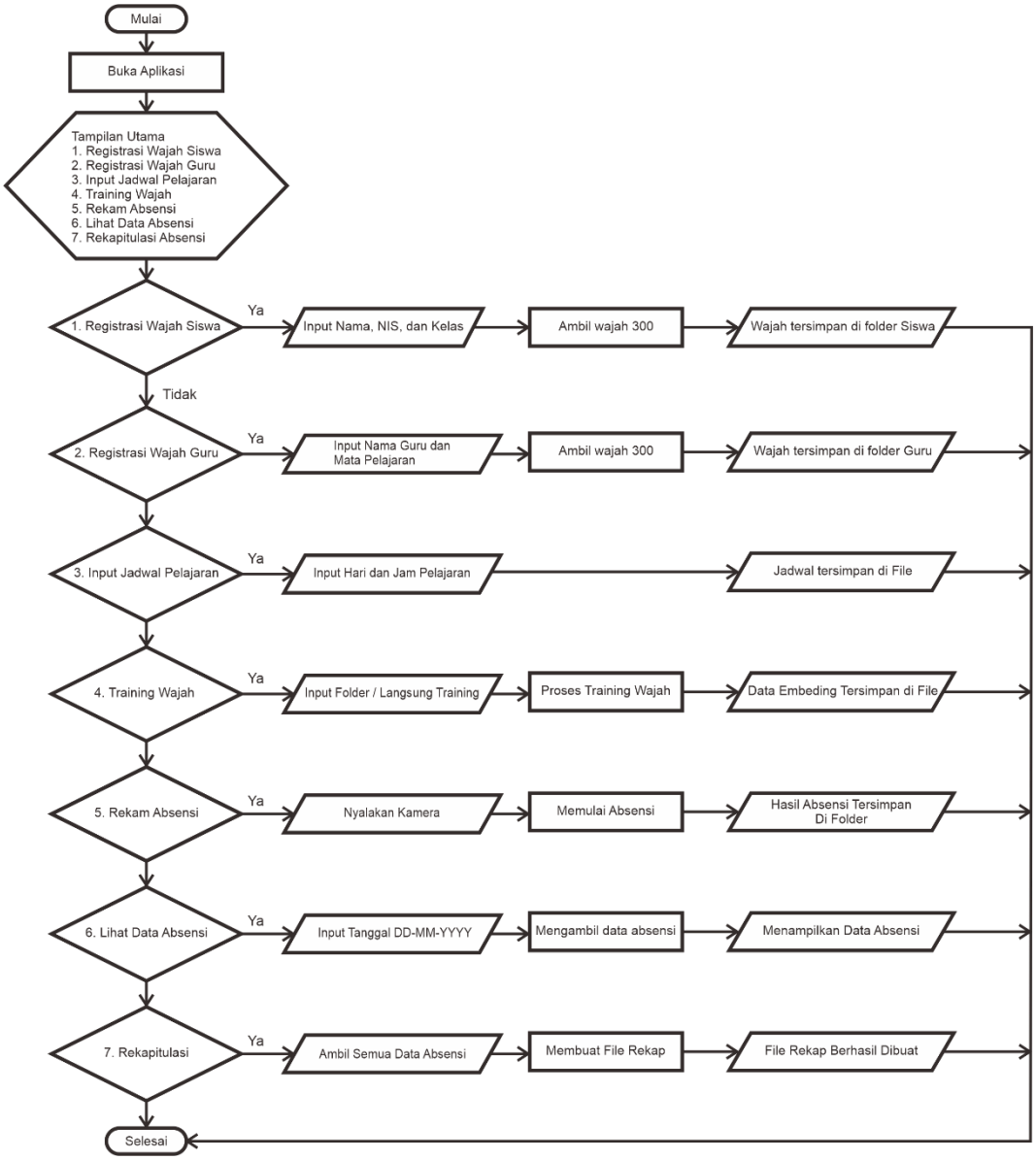
ABSENSI SISWA SMK MUHAMMADIYAH PAREPARE

Tahun Ajaran : 2023/2024
Semester : Genap
Nama Guru : Nurhidayani, S.Kom
Mata Pelajaran : Dasar - Dasar Program Keahlian TJKT
Jam Pelajaran : 08:00
Kelas : X TKJ

NO	NIS	NAMA SISWA	JK	2025-04-21	2025-04-22	Jumlah Kehadiran
1	241388	Adhe Andriansyah	L	✓	✓	2
2	241390	Farel B	L	✓	×	1
3	241391	Khaerunnisaa Biqalbi	P	✓	✓	2
4	241393	Muh. Pasha Syaputra	L	✓	✓	2
5	241395	Nuraniza Ramadani	P	✓	✓	2
6	241397	Selvi	P	✓	✓	2
7	241398	Syaiful Rahmat Ilyas	L	✓	✓	2
8	241399	Andi Nurmanani	P	✓	✓	2
9	241399	Tiara Husnul Khatimah D	P	✓	✓	2
		PARAF GURU		✓	✓	

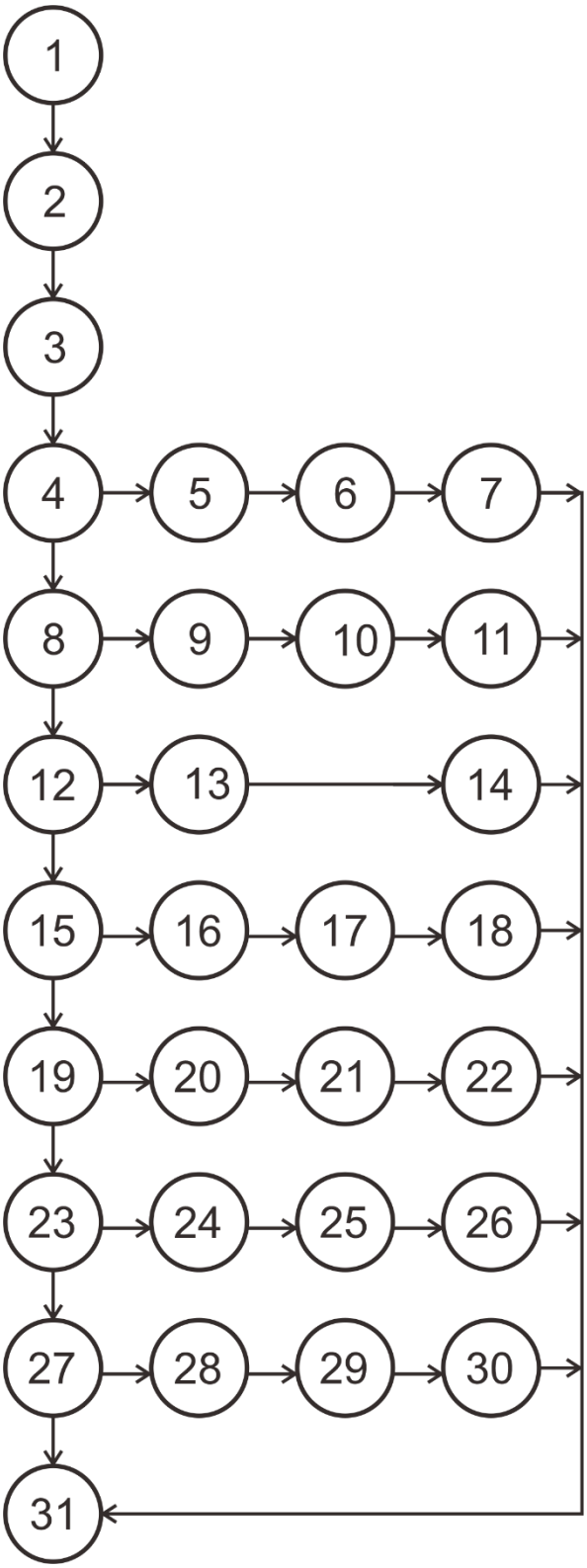
2. Pengujian whitebox

a. Flowchart pengujian white box



Gambar 4.27 Flowchart pengujian white box

b. Flowgraph pengujian Whitebox



Gambar 4.28 Flowgraph pengujian Whitebox

1) Menghitung cyclomatic complexity V(G) pada edge dan node.

   Pada rumus V(G)=E−N+2P

E (edge) = 37

N (node) = 31

P (Predikat node) = 1

Penyelesaian =

$V(G)=E-N+2P=37-31+2(1)=8$

2) Dari perhitungan cyclomatic complexity flowgraph di atas memiliki region = 8
3) Independent path pada Flowgraph tersebut yakni :

   Path 1: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 31

   Path 2: 1 → 2 → 3 → 4 → 8 → 9 → 10 → 11 → 31

   Path 3: 1 → 2 → 3 → 4 → 8 → 12 → 13 → 14 → 31

   Path 4: 1 → 2 → 3 → 4 → 8 → 12 → 15 → 16 → 17 → 18 → 31

   Path 5 : 1 → 2 → 3 → 4 → 8 → 12 → 15 → 19 → 20 → 21 → 22 → 31

   Path 6 : 1 → 2 → 3 → 4 → 8 → 12 → 15 → 19 → 23 → 24 → 25 → 26 → 31

   Path 7 : 1 → 2 → 3 → 4 → 8 → 12 → 15 → 19 → 23 → 27 → 28 → 29 → 30 → 31

   Path 8 : 1 → 2 → 3 → 4 → 8 → 12 → 15 → 19 → 23 → 27 → 31

4) Grafik matriks

Tabel 4.3 Grafik Matrik

		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	E -1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
4	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2-1=1
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
8	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2-1=1
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2-1=1
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2-1=1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	2-1=1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1-1=0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1-1=0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	2-1=1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1-1=0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1-1=0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2-1=1
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1-1=0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-1=0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0-1=-1
SUM(E+1)																																0+1=1

BAB V

PENUTUP

A. Kesimpulan

Aplikasi absensi berbasis pengenalan wajah yang dikembangkan berhasil mengintegrasikan teknologi YOLOv8 untuk mendeteksi wajah secara real-time, serta DeepFace untuk melakukan proses verifikasi identitas melalui perhitungan embedding wajah. Sistem ini mampu mencatat data absensi secara otomatis, mencakup informasi seperti nama siswa, mata pelajaran, nama guru, waktu, dan foto absensi, yang semuanya tersimpan dalam format Excel. Antarmuka yang dibangun dengan PyQt5 memudahkan pengguna dalam melakukan proses absensi, melihat data, serta menghapus entri yang tidak diperlukan.

Penggunaan 300 gambar wajah per siswa dalam tahap registrasi bertujuan untuk memastikan keberagaman sudut pandang, ekspresi, dan kondisi pencahayaan, sehingga model dapat mengenali wajah secara lebih akurat dan robust. Sementara itu, nilai threshold 0.7 pada perhitungan kesamaan embedding wajah dipilih sebagai batas optimal yang seimbang antara menghindari kesalahan identifikasi (false positive) dan kegagalan pengenalan (false negative). Tingkat akurasi pada aplikasi harus di atas 70% dari model embedding yang telah dilatih.

Penelitian ini dilakukan di SMK Muhammadiyah Kota Parepare dengan uji coba absensi selama 5 hari, dan hasilnya menunjukkan bahwa sistem mampu memberikan performa yang baik dalam proses identifikasi wajah dan pencatatan absensi secara otomatis. Oleh karena itu, aplikasi ini layak dijadikan solusi modern yang efisien dan akurat untuk mendukung manajemen kehadiran di lingkungan pendidikan

B. Saran

Untuk pengembangan lebih lanjut, disarankan agar sistem dilengkapi dengan peningkatan akurasi pengenalan wajah melalui penambahan variasi data pelatihan dengan

pencahayaan dan sudut yang berbeda. Selain itu, sistem keamanan perlu ditingkatkan dengan menambahkan fitur login dan enkripsi data absensi. Fitur notifikasi dan log aktivitas juga bisa menjadi nilai tambah agar sistem dapat memberikan informasi kehadiran secara langsung ke pihak terkait. Penggunaan database seperti MySQL atau SQLite sebaiknya dipertimbangkan untuk memudahkan pengelolaan data dalam skala besar. Terakhir, pengembangan aplikasi menjadi versi portabel atau berbasis web akan meningkatkan fleksibilitas penggunaan di berbagai perangkat dan lokasi.

DAFTAR PUSTAKA

Alfeus Adi Saputra. (2022). *Halaman Sampul Implementasi Face Recognition Pada Sistem Absensi Berbasis Android Menggunakan Metode Convolutional*. https://lib.mercubuana.ac.id/

Bagus Arisena, A., Rifanda, A. Y., Lestari, R. A., & Saputra, S. (2023). Perancangan Sistem Absensi Pada SD Islam Durrotul Hikmah Menggunakan Scan QR Code. *TEKNOBIS : Jurnal Teknologi, Bisnis Dan Pendidikan*, *01*(01), 217–223. https://jurnalmahasiswa.com/index.php/teknobis.

Fachrul, M., Fajar, S., Ramadhan, W., & Trianto, J. (n.d.). *Aplikasi Katalog Produk Berbasis Website Penerbit Cv. Eureka Media Aksara*.

Efendi, J. (2021). Black-Box Testing : Analisis Kualitas Aplikasi Source Code Bank Programming. Jurnal Teknologi Informasi dan Komunikasi), 5(1), 2021. https://doi.org/10.35870/jti

Hariani, R., & Fadillah, N. (2019). Deteksi Kehadiran Mahasiswa Secara Realtime Menggunakan Webcam dengan metode Viola Jones. InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan), 3(2), 151–154. https://doi.org/10.30743/infotekjar.v3i2.1030.

Hidayatulloh, M. S. (2021). Sistem Pengenalan Wajah Menggunakan Metode Yolo (You Only Look Once). *Universitas Medika*, i–43.

Karlina, O. E., & Indarti, D. (2019). Pengenalan Objek Makanan Cepat Saji Pada Video Dan Real Time Webcam Menggunakan Metode You Look Only Once (Yolo). Jurnal Ilmiah Informatika Komputer, 24(3), 199–208. https://doi.org/10.35760/ik.2019.v24i3.2362

Kalyani Jeslyn Lim, Clement Nathanael, Felicia Angel Wijaya, Jeson Adhi Dharma, Thaddeus Kendrick Andrian, Wilsen Soetresno, & Rahmi Yulia Ningsih. (2023). Penggunaan Bahasa Pemrograman Python Untuk Memvisualisasikan Data Peluang Selamat Dari Kecelakaan Titanic. *Jurnal Publikasi Teknik Informatika*, *2*(2), 66–79. https://doi.org/10.55606/jupti.v2i2.1735.

Kus Indrani Listyoningrum, Danise Yunaini Fenida, & Nurhasan Hamidi. (2023). Inovasi Berkelanjutan dalam Bisnis: Manfaatkan Flowchart untuk Mengoptimalkan Nilai Limbah Perusahaan. *Jurnal Informasi Pengabdian Masyarakat*, *1*(4), 100–112. https://doi.org/10.47861/jipm-nalanda.v1i4.552

Mansur, M.C.D., I.C. Miyahira, J.O. Arruda, T.N. Antoniazzi & D.M. Pimpão. 2020. Key to Unionida. [in] Damborenea, Rogers & Thorp, eds. Freshwater Invertebrates — Keys to Neotropical and Antarctic Fauna 5: 409-430.

Rakasiwi, S., Kusumo, H., & Cahyo Pangestu, A. (2022). Sistem Presensi Karyawan Menggunakan Raspberry Dengan Sensor Fingerprint Dan Webcam. Jurnal Teknik Informatika Dan Teknologi Informasi, 2(2), 75–83. https://doi.org/10.55606/jutiti.v2i2.372.

Sarosa, M., & Muna, N. (2021). Implementasi Algoritma You Only Look Once (Yolo) Untuk Implementation of You Only Look Once (Yolo) Algorithm for. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(4), 787–792. https://doi.org/10.25126/jtiik.202184407.

Sinta, (2020). Apa Itu Webcam, Fungsi, Jenis-Jenis Webcam.

Susim, T., & Darujati, C. (2021). Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV. *Jurnal Syntax Admiration*, *2*(3), 534–545. https://doi.org/10.46799/jsa.v2i3.202.

Zainal, M., Suwardoyo, U., & Arfah, A. A. (2021). *DAN FINGER PRINT BERBASIS ANDROID*. *1*(3), 1–8.