

PROJECT 1 ALJABAR LINIER
FACE RECOGNITION MENGGUNAKAN RUMUS EIGEN



DISUSUN OLEH

Syafiq Nafil Arkan.	(L0124143)
Wantech Arofiq Huda F.	(L0124144)
Naufal Farrel Budianto.	(L0124152)
Odyssey Wibi Pradana.	(L0124153)

DOSEN :

Drs. Bambang Harjito, M.App.Sc, PhD

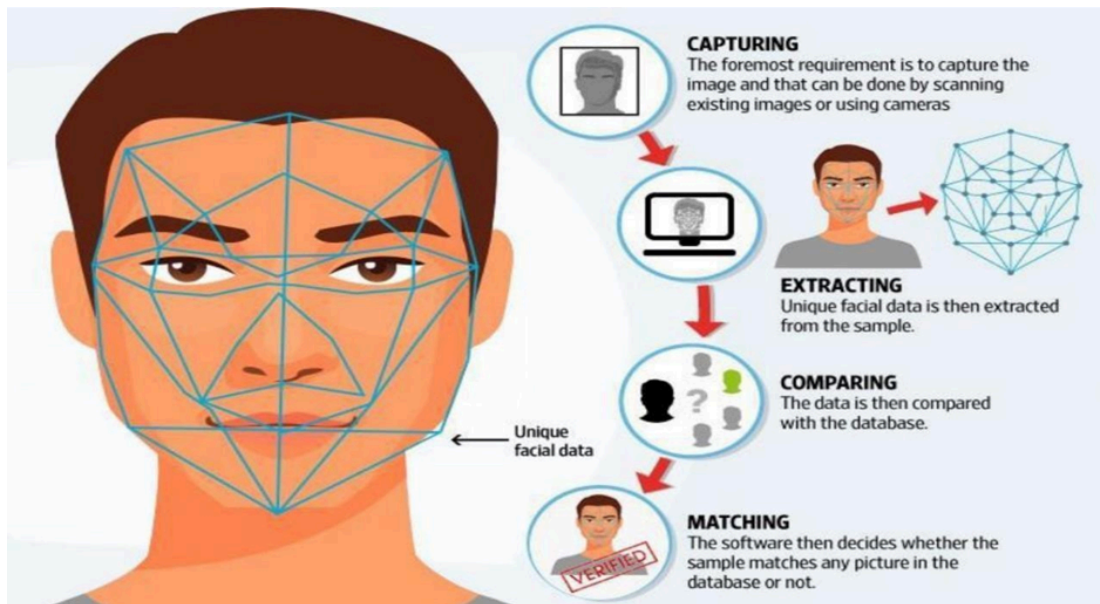
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET
2025

BAB I

DESKRIPSI MASALAH

1.1 Abstraksi

Pengenalan wajah (Face Recognition) adalah teknologi biometrik yang bisa dipakai untuk mengidentifikasi wajah seseorang untuk berbagai kepentingan khususnya keamanan. Program pengenalan wajah melibatkan kumpulan citra wajah yang sudah disimpan pada database lalu berdasarkan kumpulan citra wajah tersebut, program dapat mempelajari bentuk wajah lalu mencocokkan antara kumpulan citra wajah yang sudah dipelajari dengan citra yang akan diidentifikasi. Alur proses sebuah sistem pengenalan wajah diperlihatkan pada Gambar 1.



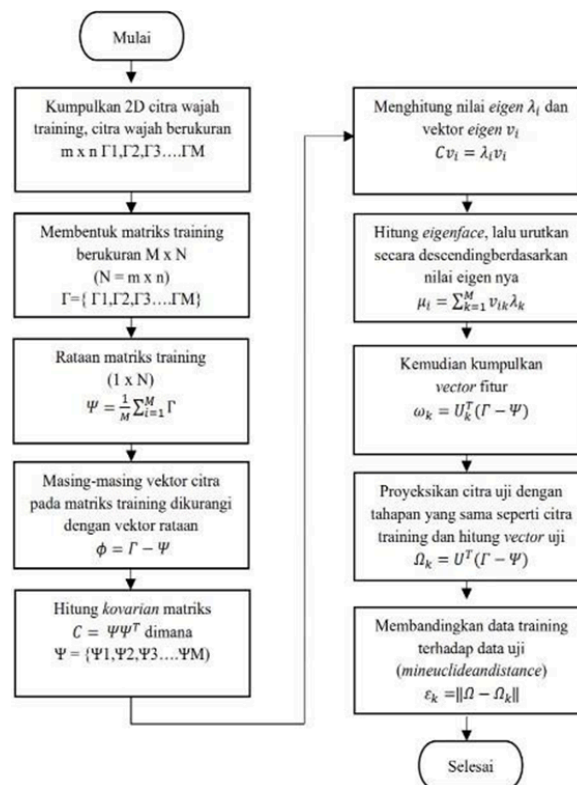
Gambar 1. Alur proses di dalam sistem pengenalan wajah (Sumber: <https://www.shadowsystem.com/page/20>)

Terdapat berbagai teknik untuk memeriksa citra wajah dari kumpulan citra yang sudah diketahui seperti jarak Euclidean dan cosine similarity, principal component analysis (PCA), serta Eigenface. Pada Tugas ini, akan dibuat sebuah program pengenalan wajah menggunakan Eigenface.

Sekumpulan citra wajah akan digunakan dengan representasi matriks. Dari

representasi matriks tersebut akan dihitung sebuah matriks Eigenface. Program pengenalan wajah dapat dibagi menjadi 2 tahap berbeda yaitu tahap training dan pencocokkan. Pada tahap training, akan diberikan kumpulan data set berupa citra wajah. Citra wajah tersebut akan dinormalisasi dari RGB ke Grayscale (matriks), hasil normalisasi akan digunakan dalam perhitungan eigenface. Seperti namanya, matriks eigenface menggunakan eigenvector dalam pembentukannya. Berikut merupakan langkah rinci dalam pembentukan eigenface

Flow chart algoritma Eigen Face



Pada tahapan akhir, akan ditemui gambar dengan euclidean distance paling kecil maka gambar tersebut yang dikenali oleh program paling menyerupai test face selama nilai kemiripan di bawah suatu nilai batas. Jika nilai minimum di atas nilai batas maka dapat dikatakan tidak terdapat citra wajah yang mirip dengan test face.

Program dibuat dengan Bahasa Python dengan memanfaatkan sejumlah library di OpenCV (Computer Vision) atau library pemrosesan gambar lainnya (contoh PIL).

Fungsi untuk mengekstraksi fitur dari sebuah citra wajah tidak perlu anda buat lagi, tetapi menggunakan fungsi ekstraksi yang sudah tersedia di dalam library. Fungsi Eigen dilarang import dari library

4 dan harus diimplementasikan, sedangkan untuk operasi matriks lainnya silahkan menggunakan library.

Kode program untuk ekstraksi fitur dapat dibaca pada artikel ini: Feature extraction and similar image search with OpenCV for newbies, pada laman: <https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774>

Nilai batas kemiripan citra test face dapat ditentukan oleh pembuat program melalui percobaan. Berikut merupakan referensi pengenalan wajah dengan metode eigenface:

<https://jurnal.untan.ac.id/index.php/jcskommipa/article/download/9727/9500>

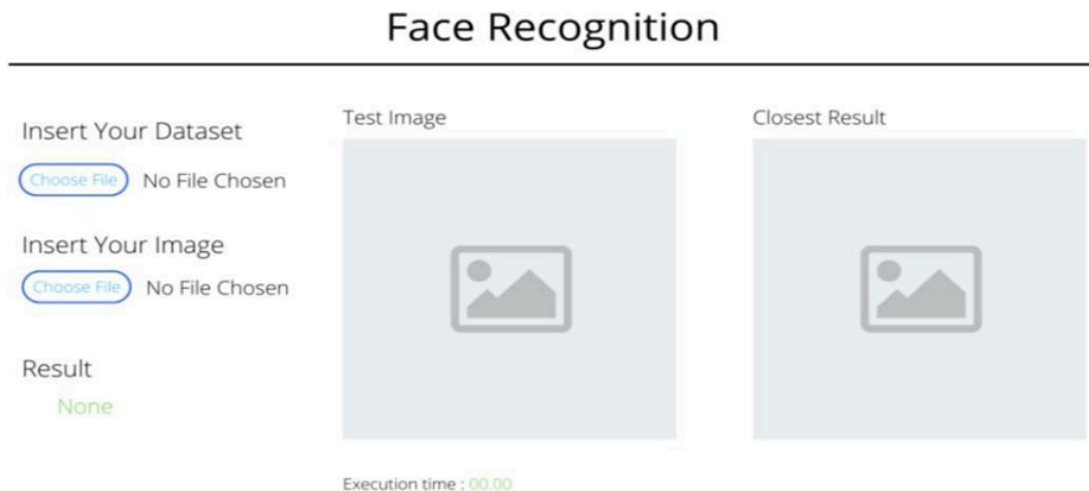
<https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm>

PENGUNAAN PROGRAM

Berikut ini adalah input yang akan dimasukkan pengguna untuk eksekusi pr

1. Folder DataSet, berisi folder atau directory yang berisi kumpulan gambar yang digunakan sebagai training image.
2. File Gambar, berisi file gambar input yang ingin dikenali dengan format file yang bebas selama merupakan format untuk gambar.

Tampilan layout dari aplikasi web yang akan dibangun kurang lebih adalah sebagai berikut. Anda dapat mengubah layout selama layout masih terdiri dari komponen yang sama.



Catatan: Warna biru menunjukkan komponen yang dapat di klik. Warna hijau menunjukkan luaran yang didapat dari hasil eksekusi. Anda dapat menambahkan menu lainnya, gambar, logo, dan sebagainya. Tampilan GUI dibuat semenarik mungkin selama mencakup seluruh informasi pada layout yang diberikan di atas. Kreativitas menjadi salah satu komponen penilaian.

1.2 Spesifikasi Tugas

Buatlah program pengenalan wajah dalam Bahasa Python berbasis GUI dengan spesifikasi sebagai berikut:

1. Program menerima input folder dataset dan sebuah gambar citra wajah.
2. Basis data wajah dapat diunduh secara mandiri melalui <https://www.kaggle.com/datasets/hereisburak/pins-face-recognition> .
3. Program menampilkan gambar citra wajah yang dipilih oleh pengguna.
4. Program melakukan pencocokan wajah dengan koleksi wajah yang ada di folder yang telah dipilih. Metrik untuk pengukuran kemiripan menggunakan eigenface + jarak euclidean.

5. Program menampilkan 1 hasil pencocokan pada dataset yang paling dekat dengan gambar input atau memberikan pesan jika tidak didapatkan hasil yang sesuai.
6. Program menghitung jarak euclidean dan nilai eigen & vektor eigen yang ditulis sendiri. Tidak boleh menggunakan fungsi yang sudah tersedia di dalam library atau Bahasa Python.

BAB II

TEORI SINGKAT MENGENAI PERKALIAN MATRIKS, NILAI EIGEN, VEKTOR EIGEN, DAN EIGENFACE

2.1 Pengenalan Wajah (Face Recognition)

Pengenalan wajah adalah salah satu teknologi biometrik yang digunakan untuk keamanan data. Metode algoritma pengenalan wajah sudah banyak diimplementasikan dan berhasil menciptakan keakuratan yang tinggi. Salah satunya penggunaan algoritma pengenalan wajah yaitu algoritma eigenface yang diekstraksi melalui principle component analysis (PCA) dilaporkan dapat menghasilkan tingkat akurasi pengenalan wajah sebesar 90.83% (Saefullah, 2015).

2.2 Perkalian Matriks

Matriks merupakan himpunan elemen-elemen bilangan yang disusun dalam bentuk baris dan kolom. Matriks banyak digunakan dalam berbagai bidang ilmu, termasuk dalam pengolahan citra digital dan machine learning. Operasi dasar dalam matriks terdiri dari penjumlahan, pengurangan, perkalian, dan invers matriks. Dua jenis perkalian yang umum digunakan adalah:

1. Perkalian Matriks dengan Skalar, yaitu setiap elemen dalam matriks dikalikan dengan suatu konstanta (skalar).
2. Perkalian Matriks dengan Matriks, yaitu operasi antara dua matriks A berukuran $m \times n$ dan B berukuran $n \times r$ menghasilkan matriks baru C berukuran $m \times r$, dengan rumus:

$$C_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$$

Operasi perkalian matriks, baik antar matriks maupun dengan skalar, merupakan dasar penting dalam perhitungan transformasi linear, termasuk pada algoritma PCA dan Eigenface (Strang, 2016).

2.3 Nilai Eigen

Nilai eigen adalah nilai karakteristik dari sebuah matriks yang berukuran $n \times n$. Nilai ini diperoleh dari operasi $Ax = \lambda x$, di mana A adalah matriks berukuran $n \times n$, x adalah vektor eigen, dan λ adalah skalar yang disebut nilai eigen. Dalam konteks pengenalan wajah, nilai eigen sangat penting karena dapat merepresentasikan fitur-fitur penting dari wajah dalam bentuk numerik yang ringkas namun informatif. Dengan memanfaatkan nilai eigen, sistem dapat melakukan ekstraksi ciri-ciri wajah secara efisien, sehingga memperkuat proses identifikasi dan meningkatkan akurasi pengenalan wajah (Turk, 1991).

2.4 Vektor Eigen

Vektor eigen adalah vektor tidak nol di \mathbb{R}^n yang dihasilkan dari operasi yang sama seperti pada nilai eigen. Vektor ini menyatakan arah dari transformasi matriks terhadap data, di mana ketika vektor eigen dikalikan dengan matriks A , hasilnya adalah kelipatan dari vektor itu sendiri. Dalam pengenalan wajah, vektor eigen digunakan untuk membentuk eigenface atau representasi wajah-wajah dalam ruang fitur. Dengan mengubah data wajah menjadi kombinasi dari beberapa vektor eigen utama, sistem dapat membedakan antara wajah yang satu dengan yang lain secara lebih akurat dan efisien (Turk, 1991).

2.5 Eigenface

Eigenface adalah metode pengenalan wajah yang memanfaatkan teknik Principal Component Analysis (PCA) untuk mereduksi dimensi data citra wajah. Dengan PCA, citra wajah yang kompleks dapat direpresentasikan dalam bentuk vektor dengan dimensi lebih rendah, yang disebut sebagai eigenface. Setiap eigenface merupakan kombinasi linier dari citra wajah dalam basis data pelatihan, yang menangkap variasi utama di antara wajah-wajah tersebut.

2.6 Euclidean Distance

Jarak Euclidean adalah metode untuk mengukur jarak terpendek antara dua titik dalam ruang berdimensi m . Dalam pengenalan wajah berbasis eigenface, metode ini digunakan untuk menghitung kemiripan antara citra uji dan citra pelatihan yang sudah direpresentasikan sebagai vektor.

Rumus jarak Euclidean antara dua vektor x dan y adalah

$$d(x,y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

Semakin kecil nilai $d(x,y)$, maka semakin mirip kedua citra tersebut. Karena kesederhanaan dan efisiensinya, metode ini banyak digunakan dalam sistem klasifikasi berbasis PCA dan Eigenface (Widodo, 2013)

BAB 3

IMPLEMENTASI PROGRAM

Pada bab ini dijelaskan implementasi program sistem pengenalan wajah berbasis metode *Eigenface* yang dibangun menggunakan bahasa pemrograman Python dengan antarmuka web berbasis Flask. Sistem dirancang agar pengguna dapat mengunggah dataset wajah dalam format ZIP serta gambar wajah input, kemudian sistem memproses dan mencocokkan gambar tersebut berdasarkan fitur wajah.

3.1 Struktur Direktori Program

Struktur direktori proyek terdiri dari beberapa komponen penting sebagai berikut:

```
face_recognition_app/
|
|— app.py           # Aplikasi Flask utama
|— face_recognition.py # Modul logika pengenalan wajah
|— static/
|   |— uploads/      # Folder penyimpanan dataset dan hasil pemrosesan
|— templates/
|   |— index.html     # Template HTML utama
|— requirements.txt   # Daftar dependensi Python
```

3.2 Penjelasan Komponen Program

3.2.1 **app.py**

File ini berisi logika utama backend yang menggunakan Flask. Fungsionalitas penting antara lain:

- **Route "/"**: Menyediakan halaman awal dengan formulir unggah file.
- **Route "/recognize"**: Memproses unggahan file, mengekstrak dataset ZIP, menyimpan

input gambar, lalu memanggil fungsi `run_recognition()` dari `face_recognition.py`. Hasilnya kemudian ditampilkan kembali di halaman utama.

3.2.2 `face_recognition.py`

File ini merupakan inti dari logika pemrosesan wajah menggunakan metode Eigenface. Proses utamanya meliputi:

1. **Load Dataset:** Membaca seluruh gambar di dalam folder dataset, mengubah menjadi grayscale, resize ke ukuran 100x100, lalu flatten menjadi vektor.
2. **Hitung Mean dan Kovarians:** Menghitung wajah rata-rata dan matriks kovarians sebagai dasar ekstraksi fitur.
3. **Eigen Decomposition:** Menggunakan algoritma manual berbasis *power iteration* dan *deflation* untuk mendapatkan *eigenfaces*.
4. **Proyeksi Wajah:** Semua wajah dan input diproyeksikan ke ruang Eigenface.
5. **Pencocokan Wajah:** Menggunakan jarak Euclidean untuk menentukan kecocokan antara wajah input dan dataset.

3.3 Alur Kerja Sistem

1. Pengguna mengunggah:
 - **Dataset wajah** dalam bentuk file ZIP (berisi gambar-gambar wajah).
 - **Satu gambar input** untuk dikenali.

2. Sistem:

- Membuat folder sesi berdasarkan UUID.
- Mengekstrak ZIP ke folder `dataset/`.
- Menyimpan gambar input sebagai `input.jpg`.
- Memanggil fungsi `run_recognition()` untuk mencocokkan gambar input terhadap dataset.
- Menyimpan hasil gambar `matched.jpg`.

3. Hasil ditampilkan ke pengguna berupa:

- Label atau nama file gambar yang paling cocok.
- Waktu proses.
- Tampilan visual dari gambar input dan gambar hasil pencocokan.

3.4 Tampilan Antarmuka

Antarmuka dibangun menggunakan HTML melalui template Flask (`index.html`). Pengguna dapat mengunggah file dan melihat hasil pencocokan wajah secara langsung.

Contoh tampilan halaman:

- Formulir unggah dataset dan gambar input.
- Gambar input dan gambar hasil pencocokan ditampilkan berdampingan.

- Informasi label gambar cocok dan waktu proses ditampilkan di bawah gambar.

3.5 Teknologi yang Digunakan

Komponen	Teknologi yang Digunakan
Bahasa	Python 3.x
Framework	Flask
Library Citra	OpenCV
Numerik	NumPy
Antarmuka	HTML (Jinja2 Template Flask)
File Handling	zipfile, os, uuid

3.6 Penanganan Kesalahan (Error Handling)

Beberapa penanganan kesalahan telah diimplementasikan agar sistem tidak crash saat terjadi error, seperti:

- Validasi keberadaan file input.
- Pemeriksaan format file.
- Validasi struktur file ZIP.
- Penanganan error pemrosesan gambar.

BAB 4

EKSPERIMEN

Bab ini menyajikan hasil eksperimen yang dilakukan terhadap sistem pengenalan wajah yang telah diimplementasikan. Eksperimen dilakukan menggunakan beberapa contoh dataset dan gambar input untuk menguji keakuratan dan performa sistem dalam mengenali wajah berdasarkan metode *Eigenface*. Selain itu, dilakukan analisis terhadap hasil eksekusi tersebut untuk menilai efektivitas pendekatan yang digunakan.

4.1 Prosedur Eksperimen

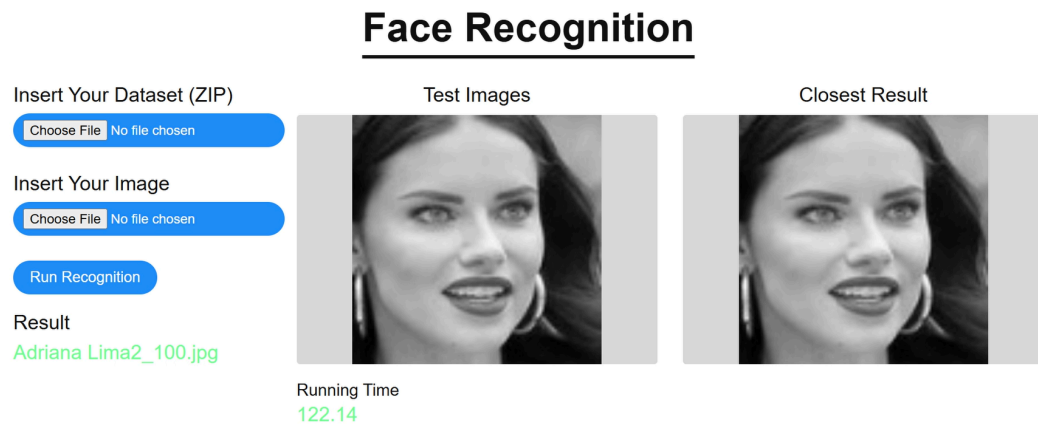
Eksperimen dilakukan dengan tahapan sebagai berikut:

1. Menyiapkan dataset wajah yang berisi lebih dari 200 file gambar wajah, dengan sudut pandang dan pencahayaan yang berbeda beda.
2. Gambar-gambar dalam dataset disimpan dalam folder dan dikompres dalam format ZIP untuk diuji di sistem, hal dilakukan karena Graphical User Interface (GUI) program menggunakan html, yang dimana pada html tidak dapat mengupload sebuah folder secara langsung, namun harus dikompresi kedalam file ZIP atau RAR.
3. Menyediakan gambar input dari salah satu orang yang sudah ada dalam dataset (uji positif) dan gambar wajah orang lain (uji negatif).
4. Mengamati hasil pencocokan dan mencatat waktu proses yang dibutuhkan.

Semua gambar diubah ke grayscale dan diresize ke ukuran 100×100 piksel sebelum diproses.

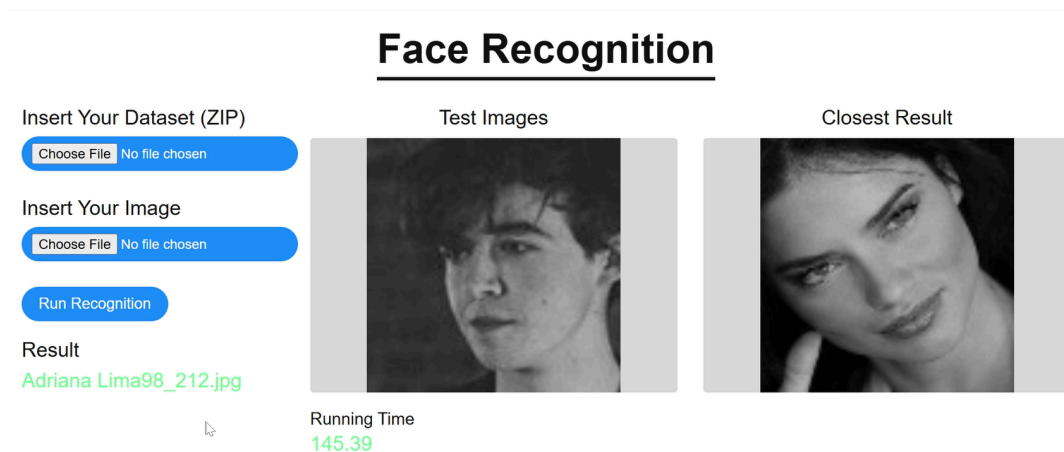
4.2 Hasil Eksekusi Program

A. Apabila Gambar Yang Dijadikan Test Image Terdapat Dalam Folder Dataset



Pada eksperimen ini, di dalam folder dataset training image kami terdapat foto yang sama persis dengan foto yang menjadi test image, oleh karena itu foto yang keluar juga merupakan foto yang sama dengan foto test image. Folder training image yang kami gunakan memiliki 213 gambar di dalamnya dan membutuhkan waktu 122,14 detik untuk eksekusi program

B. Apabila Gambar Yang Dijadikan Test Image Tidak Ada Dalam Folder Dataset



Pada eksperimen ini, di dalam folder dataset training image tidak terdapat foto yang sama dengan foto training image, namun program akan mencari foto paling mirip dengan foto pada test image. Dapat dilihat program kami akan mengeluarkan foto orang yang terdeteksi paling mirip dengan test image. Folder training image yang kami gunakan memiliki 213 image di dalamnya dan membutuhkan waktu sekitar 145,39 detik untuk eksekusi program.

BAB 5

KESIMPULAN

5.1 Kesimpulan

Dalam proses pengenalan wajah (face recognition), terdapat berbagai metode yang dapat diterapkan, salah satunya adalah algoritma eigenface. Algoritma ini mampu mengenali wajah dengan memanfaatkan variasi-variasi dari wajah yang terdapat dalam kumpulan gambar, serta menggunakan variasi tersebut untuk melakukan proses enkripsi dan dekripsi wajah.

Pada tugas besar ini, kami merancang sebuah program pengenalan wajah yang menggunakan algoritma eigenface sebagai dasar. Program ini kami implementasikan dalam bentuk antarmuka grafis (GUI) menggunakan library streamlit. Hasilnya, program kami mampu mengenali gambar wajah dari individu yang sama dengan tingkat akurasi yang baik dan waktu eksekusi yang normal.

5.2 Saran

Saran untuk kelompok kami di antaranya:

1. Optimasi kode program agar running time-nya lebih cepat, misalnya dengan menggunakan algoritma yang lebih efisien atau menghindari proses yang redundan.
2. Pemberian komentar pada kode program sebaiknya lebih jelas dan konsisten, sehingga proses debugging dan pemahaman alur kode menjadi lebih mudah.

5.3 Refleksi

Dengan terselesaikannya tugas besar kedua ini, kami memperoleh banyak manfaat dan pengalaman baru. Kami belajar tentang algoritma pengenalan wajah menggunakan OpenCV Python, yang sebelumnya belum pernah kami pelajari. Selain meningkatkan

kemampuan teknis dalam pemrograman, kami juga mengasah berbagai soft skill seperti kerja sama tim, manajemen waktu, dan problem solving. Kami juga mulai menyadari pentingnya mengoptimalkan running time dalam pengembangan program agar hasilnya lebih efisien. Meskipun begitu, manajemen waktu masih menjadi tantangan bagi kami karena sebagian besar pengerjaan dilakukan mendekati deadline.

DAFTAR PUSTAKA

- [1] S. P. G. Aradhya, N. R. K. Kumar, and M. A. H. Khan, "Implementation of Eigenface Based Face Recognition on FPGA Using High Level Synthesis," dalam *2024 International Conference for Advancement in Technology (ICONAT)*, 2024. doi: 10.1109/ICONAT61821.2024.10614584.
- [2] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, hlm. 71–86, 1991.
- [3] A. Kristiadi, "Feature Extraction and Similar Image Search with OpenCV for Newbies," *Medium*, 21 November 2017. [Online]. Tersedia: <https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774>. [Diakses: 11 Juni 2025].
- [4] A. Rosebrock, "OpenCV Eigenfaces for Face Recognition," *PyImageSearch*, 10 Mei 2021. [Online]. Tersedia: <https://pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/>. [Diakses: 11 Juni 2025].
- [5] Royi, "Answer to 'Face Recognition- Simplistic Explanation on PCA Eigenface Algorithm'," *Signal Processing Stack Exchange*, 20 Desember 2016. [Online]. Tersedia: <https://dsp.stackexchange.com/questions/32017/face-recognition-simplistic-explanation-on-pca-eigenface-algorithm>. [Diakses: 11 Juni 2025].
- [6] H. I. Mansour, "Face Recognition using Eigenfaces (Python)," *Medium*, 26 Agustus 2022. [Online]. Tersedia: <https://hassan-id-mansour.medium.com/face-recognition-using-eigenfaces-python-b857b2599ed0>. [Diakses: 11 Juni 2025].
- [7] E. Ku, "Beginner-Friendly Power Iteration, Clearly Explained (Page-Rank Pt. 1)," *Medium*, 28 Maret 2020. [Online]. Tersedia: <https://medium.com/@eugeneku123/beginner-friendly-power-iteration-clearly-explained-page-rank-pt-1-8d4d371ba2cf>. [Diakses: 11 Juni 2025].

LAMPIRAN

Video Demo

<https://youtu.be/0CMJHMB6SL4>

Github Repository

<https://github.com/kacangcina/Face-Recognition.git>