

IMPLEMENTASI PENCARIAN NILAI EIGEN DAN EIGENFACE DALAM BAHASA PEMROGRAMAN **PYTHON UNTUK PENGENALAN WAJAH**

Disusun untuk memenuhi laporan tugas besar mata kuliah Aljabar Linier dan Geometri semester 3 di Institut Teknologi Bandung.



Disusun oleh Kelompok 28 “Double Hoki”:

Azmi Hasna Zahrani (13521006)

Laila Bilbina Khairu Nisa (13521016)

Ryan Samuel Chandra (13521140)

**TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

Jl. Ganesh No. 10, Lb. Siliwangi, Kecamatan Coblong,
Kota Bandung, Jawa Barat, 40132

2022 / 2023

PRAKATA

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat-Nya begitu melimpah dalam kehidupan kami. Akhirnya, kami dapat menyelesaikan laporan ini tepat waktu setelah 3 minggu penuh melakukan kerja sama pemrograman secara rutin. Kami juga mengucapkan terima kasih kepada pihak ITB, khususnya Bapak Dr. Ir. Rinaldi Munir, M.T., serta seluruh kakak-kakak asisten mata kuliah Aljabar Linier dan Geometri, yang telah memberikan kami kesempatan untuk mengerjakan tugas ini.

Berikut disajikan laporan lengkap tentang “Implementasi Pencarian Nilai Eigen dan *Eigenface* dalam Bahasa Pemrograman *Python* untuk Pengenalan Wajah”. Laporan ini selain dibuat agar dapat bermanfaat bagi masyarakat, juga secara khusus disusun untuk memenuhi salah satu tugas besar dalam mata kuliah Aljabar Linier dan Geometri di semester 3 Teknik Informatika Institut Teknologi Bandung (IF ITB).

Dengan semua informasi yang terkumpul dari berbagai sumber serta percobaan, kami berusaha sebaik mungkin untuk memenuhi tujuan pembuatan yang telah ditetapkan di tengah hambatan lingkungan dan tekanan waktu.

Kami berharap laporan ini dapat menjadi sesuatu yang memuaskan bagi semua pembaca. Tetapi kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Untuk itu, kami mohon kritik dan saran yang bersifat konstruktif untuk perkembangan percobaan kami.

Bandung, 22 November 2022

Penyusun Laporan

Daftar Isi

Prakata	iii
Daftar Isi	iv
BAB 1 : PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Deskripsi Permasalahan	1
BAB 2 : TINJAUAN PUSTAKA	3
2.1 Perkalian Matriks	3
2.2 Nilai Eigen	4
2.3 <i>Eigenface</i>	5
BAB 3 : IMPLEMENTASI PROGRAM DALAM BAHASA PYTHON	7
3.1 <i>Library</i> Buatan untuk Melengkapi <i>Library</i> Bawaan	7
3.2 <i>Mean Face</i>	8
3.3 Keberjalan Program Utama	8
3.4 Pengaplikasian Fitur Bonus	8
BAB 4 : EKSPERIMEN	9
4.1 Mengeluarkan Foto yang Valid	9
4.2 Keluaran dari Masukan yang Tidak Valid	10
4.3 Inputan Di Kamera (program bonus)	11
BAB 5 : PENUTUP	13
5.1 Kesimpulan	13
5.2 Saran	13
5.3 Refleksi	13
DAFTAR REFERENSI	15
LAMPIRAN	15

BAB 1

PENDAHULUAN

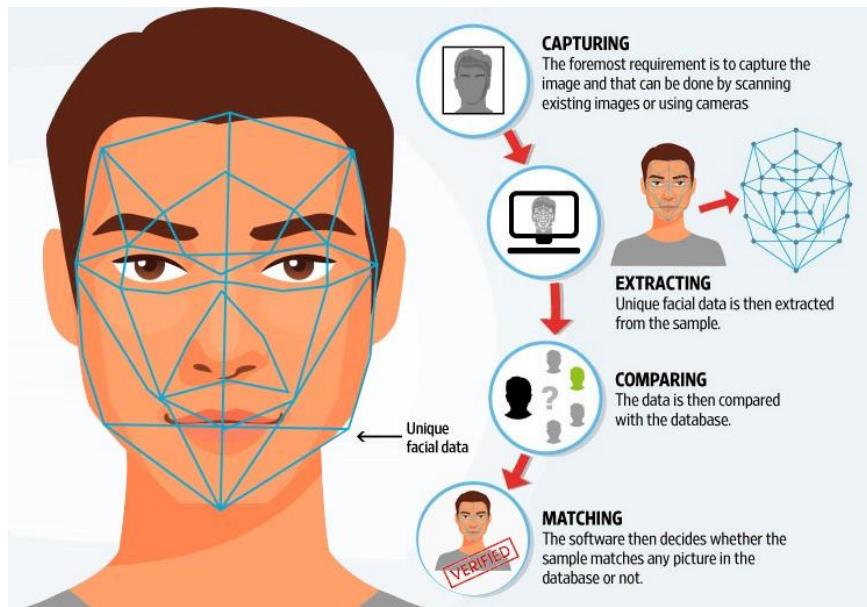
1.1 Latar Belakang

Sebagai manusia, kita diciptakan oleh Yang Mahakuasa dengan dilengkapi oleh akal budi. Inilah yang membuat manusia berbeda dengan ciptaan lainnya, karena manusia bisa berpikir untuk membuat keputusan, mencari solusi, dan membangun peradaban. Meskipun kita tidak mungkin mengetahui keseluruhan isi dunia dengan pasti, sejauh ini ilmu pengetahuan telah berkembang dengan sangat pesat. Berbagai teori dan penemuan telah diciptakan oleh para ahli dengan latar belakang yang berbeda-beda, yang secara intensif mempelajari tentang alam semesta dari berbagai sisi. Hal itu berdampak pada munculnya banyak penelitian untuk mengembangkan beragam hal yang pernah, atau bahkan belum pernah dikemukakan.

Salah satu ilmu yang sedang mengalami proses pengembangan secara pesat adalah ilmu komputer. Di zaman modern, semua kegiatan di dunia tidak terlepas dari peranan teknologi digital, yang sejatinya berakar dari program komputer. Di dalamnya terdapat berbagai macam algoritma unik, dengan tujuan membantu manusia dalam menyelesaikan tugas masing-masing. Berbicara tentang teknologi, rasanya tidak lengkap jika kita tidak mengindahkan alat pendekripsi wajah, yang telah banyak berjasa dalam dunia pendidikan maupun pekerjaan. Biasanya, alat tersebut dipakai untuk mempermudah pengabsenan serta penelitian forensik.

1.2 Deskripsi Permasalahan

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 pencocokan. Pada tahap *training*, akan diberikan kumpulan set data berupa citra wajah. Citra wajah tersebut akan dinormalisasi dari RGB ke *grayscale* (matriks). Lalu, hasil normalisasi akan digunakan dalam perhitungan *eigenface*. Seperti namanya, matriks *eigenface* menggunakan *eigenvector* dalam pembentukannya. Langkah rinci dalam pembentukan *eigenface* akan dijelaskan di bagian lain dari laporan ini.

BAB 2

TINJAUAN PUSTAKA

2.1 Perkalian Matriks

Menurut KBBI, matriks dalam dunia komputasi berarti larik dua dimensi yang disusun dalam baris dan kolom yang nilai baris dan kolomnya dapat sama atau berbeda. Maka dari itu, setiap elemen matriks terdefinisi berdasarkan dua buah indeks yang secara berurutan mengacu tepat pada sebuah baris dan sebuah kolom pada matriks. Indeks harus bertipe mempunyai suksesor dan predecesor yang pasti, misalnya *integer* (bilangan bulat).

Sebagai sebuah struktur dalam bidang komputasi sekaligus matematika, terdapat berbagai operasi yang bisa dilakukan terhadap sebuah matriks. Salah satunya adalah perkalian. Perkalian yang dimaksud terbagi menjadi dua, yaitu perkalian matriks dengan matriks, dan perkalian matriks dengan skalar.

Operasi perkalian matriks dapat dilakukan pada dua buah matriks, misalnya A dan B , jika jumlah kolom matriks A = jumlah baris matriks B . Perkalian ini menghasilkan matriks baru, misalnya C . Setiap elemen C (c_{ij}) merupakan penjumlahan dari perkalian elemen-elemen A baris ke- i dengan elemen-elemen B kolom ke- j . Contoh operasi perkalian matriks bisa dilihat pada matriks A (berukuran

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}, B = \begin{bmatrix} k & n \\ l & 0 \\ m & p \end{bmatrix}, \text{ maka}$$

2×3) dan B (berukuran 3×2) berikut ini:

$$A \times B = \begin{bmatrix} ak + bl + cm & an + bo + cp \\ dk + el + fm & dn + eo + fp \end{bmatrix}$$

Dapat disimpulkan bahwa perkalian matriks berukuran $p \times q$ dengan matriks berukuran $q \times r$ akan menghasilkan matriks berukuran $p \times r$. Perlu diperhatikan juga bahwa perkalian ini tidak bersifat komutatif, yakni $A \times B$ tidak sama dengan $B \times A$. Tidak lupa, perkalian dua buah matriks dapat dituliskan sebagai kombinasi linier dari kolom-kolomnya.

- Contoh: perkalian matriks dapat ditulis sebagai kombinasi linier

$$\begin{bmatrix} -1 & 3 & 2 \\ 1 & 2 & -3 \\ 2 & 1 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -9 \\ -3 \end{bmatrix}$$

$$2 \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} - 1 \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} + 3 \begin{bmatrix} 2 \\ -3 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 \\ -9 \\ -3 \end{bmatrix}$$

Matriks bisa dikalikan dengan sebuah skalar. Jika X adalah sebuah matriks dan c adalah skalar, maka cX menghasilkan matriks baru, misalnya Y . Masing-masing elemen Y adalah elemen X pada indeks yang sama, dikalikan dengan c . Contohnya sebagai berikut:

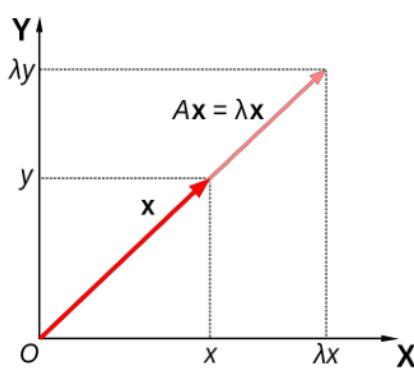
$$A = \begin{bmatrix} 2 & 3 & 4 \\ 1 & 3 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 2 & 7 \\ -1 & 3 & -5 \end{bmatrix}, \quad C = \begin{bmatrix} 9 & -6 & 3 \\ 3 & 0 & 12 \end{bmatrix}$$

$$2A = \begin{bmatrix} 4 & 6 & 8 \\ 2 & 6 & 2 \end{bmatrix}, \quad (-1)B = \begin{bmatrix} 0 & -2 & -7 \\ 1 & -3 & 5 \end{bmatrix}, \quad \frac{1}{3}C = \begin{bmatrix} 3 & -2 & 1 \\ 1 & 0 & 4 \end{bmatrix}$$

2.2 Nilai Eigen dan Vektor Eigen

Jika A adalah matriks berukuran $n \times n$, maka vektor tak-nol \mathbf{x} di \mathbb{R}^n disebut vektor eigen dari A , jika $A\mathbf{x}$ sama dengan perkalian suatu skalar λ dengan \mathbf{x} , yaitu $A\mathbf{x} = \lambda\mathbf{x}$. Skalar λ disebut nilai eigen dari A , sedangkan \mathbf{x} dinamakan vektor eigen yang berkoresponden dengan λ .

Nilai eigen menyatakan nilai karakteristik dari sebuah matriks yang bujur sangkar. Vektor eigen \mathbf{x} menyatakan matriks kolom yang apabila dikalikan dengan sebuah matriks bujur sangkar menghasilkan vektor lain yang merupakan kelipatan vektor itu sendiri.



Cara menghitung nilai eigen dan vektor eigen didapat dengan menurunkan persamaan sebelumnya seperti ini:

$$\begin{aligned} A\mathbf{x} &= \lambda\mathbf{x} \\ I\mathbf{A}\mathbf{x} &= \lambda I\mathbf{x} \\ A\mathbf{x} &= \lambda I\mathbf{x} \\ (\lambda I - A)\mathbf{x} &= 0 \end{aligned}$$

Agar persamaan tersebut memiliki solusi tidak-nol, maka haruslah $|\lambda I - A| = 0$. Ini adalah persamaan karakteristik dari matriks A . Akar-akarnya, yaitu λ dinamakan akar-akar karakteristik atau nilai eigen.

Contoh sebagai berikut: Tentukanlah nilai eigen dan vektor eigen dari matriks $A = [3 \ 0 \ 8 \ -1]$.

$$\lambda I - A = \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix} = \begin{bmatrix} \lambda - 3 & 0 \\ -8 & \lambda + 1 \end{bmatrix}$$

$$\det(\lambda I - A) = 0 \rightarrow \begin{vmatrix} \lambda - 3 & 0 \\ -8 & \lambda + 1 \end{vmatrix} = 0 \rightarrow (\lambda - 3)(\lambda + 1) = 0 \rightarrow \text{persamaan karakteristik} \\ \rightarrow \lambda_1 = 3 \text{ dan } \lambda_2 = -1$$

Jadi, nilai-nilai eigen dari matriks A adalah $\lambda = 3$ dan $\lambda = -1$.

Selanjutnya, kita hitung vektor eigen yang membentuk ruang eigen dari masing-masing nilai.

$$\text{Untuk } \lambda = 3 \rightarrow \begin{bmatrix} 0 & 0 \\ -8 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow -8x_1 + 4x_2 = 0 \rightarrow 8x_1 = 4x_2 \rightarrow x_1 = \frac{1}{2}x_2$$

$\rightarrow \text{Solusi: } x_1 = \frac{1}{2}t, x_2 = t, t \in \mathbb{R}$

$$\text{Vektor eigen: } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}t \\ t \end{bmatrix} = t \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} \rightarrow \text{membentuk ruang eigen (eigenspace)}$$

Jadi, $\begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}$ adalah basis untuk ruang eigen dengan $\lambda = 3$

$$\text{Ruang eigen ditulis sebagai } E(3) = \{ \mathbf{x} = t \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}, t \in \mathbb{R} \}$$

$$\text{Untuk } \lambda = -1 \rightarrow \begin{bmatrix} \lambda - 3 & 0 \\ -8 & \lambda + 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -4 & 0 \\ -8 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Selesaikan dengan eliminasi Gauss:

$$\begin{bmatrix} -4 & 0 & 0 \\ -8 & 0 & 0 \end{bmatrix} \xrightarrow{\text{R1}/(-4)} \begin{bmatrix} 1 & 0 & 0 \\ -8 & 0 & 0 \end{bmatrix} \xrightarrow{\text{R2} + 8\text{R1}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\rightarrow \text{Solusi: } x_1 = 0, x_2 = t, t \in \mathbb{R}$

$$\text{Vektor-vektor eigen: } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ t \end{bmatrix} = t \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \text{membentuk ruang eigen (eigenspace)}$$

Jadi, $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ adalah basis untuk ruang eigen dengan $\lambda = -1$

$$\text{Ruang eigen ditulis sebagai } E(-1) = \{ \mathbf{x} = t \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t \in \mathbb{R} \}$$

2.3 Eigenface

Secara umum, sistem pengenalan wajah dibagi menjadi 2 jenis, yaitu sistem *feature-based* dan sistem *image-based*. Pada sistem pertama, fitur lokal didapat dari proses ekstraksi komponen citra wajah seperti mulut, hidung, mata, pipi dan lain-lain. Kemudian, hubungan antar fitur-fitur tersebut dimodelkan secara geometris. Sedangkan, sistem yang kedua menggunakan informasi mentah dari *pixel* citra, yang kemudian direpresentasikan dalam metode tertentu.

Pada tahun 1991, Turk dan Pentland menyarankan pendekatan pengenalan wajah yang menggunakan reduksi dimensionalitas dan konsep aljabar linier. Pendekatan ini secara komputasi lebih murah dan mudah diimplementasikan, dan dengan demikian, digunakan dalam berbagai aplikasi pada saat itu seperti pengenalan tulisan tangan, pembacaan bibir, analisis citra medis, dan lain-lain.

Metode ini disebut *eigenface*, yang mengubah sampel/data pelatihan (*training image*) menjadi sebuah vektor *flat* (gabungan vektor) dan digabung bersama-sama menjadi sebuah matriks tunggal. Vektor eigen kemudian diekstraksi melalui berbagai operasi dan disimpan sebagai “*database*”.

PCA (*Principal Component Analysis*) adalah teknik reduksi dimensionalitas yang diusulkan oleh Pearson pada tahun 1901. Teknik inilah yang akan digunakan untuk untuk pengenalan wajah dengan *eigenface*. Seperti yang sudah disebutkan, prosesnya melibatkan nilai eigen dan vektor eigen, lalu memproyeksikan *training image* pada ruang fitur yang lebih kecil. Algoritmanya bisa dilihat lebih detail sebagai berikut:

1. Data disiapkan dengan membuat suatu himpunan S yang terdiri dari seluruh *training image*; $S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M)$.
2. Nilai rata-rata atau *mean* (Ψ) diambil dari masing-masing indeks yang bersesuaian pada seluruh gambar.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

3. Selisih (ϕ) antara nilai *training image* (Γ_i) dengan nilai tengah (Ψ) dihitung.

$$\phi_i = \Gamma_i - \Psi$$

4. Matriks kovarian (C) dihitung dengan cara berikut:

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T$$

5. Nilai eigen (λ) dan vektor eigen (v) dari matriks kovarian dihitung.

$$C \times v_i = \lambda_i \times v_i$$

6. *Eigenface* dapat dicari dengan:

$$\mu_i = \sum_{k=1}^M v_{ik} \times \phi_k$$

7. Sebuah image wajah baru atau *test face* (Γ_{new}) akan dicoba untuk dikenali. Pertama, cara pada tahapan pertama perhitungan *eigenface* diterapkan untuk mendapatkan nilai eigen dari *image* tersebut.

$$\mu_{new} = v \times \Gamma_{new} - \Psi$$

$$\Omega = \mu_1, \mu_2, \dots, \mu_M$$

8. Metode *Euclidean distance* digunakan untuk mencari jarak terpendek antara nilai eigen dari *training image* dalam *database* dengan nilai eigen dari *image test face*.

$$\varepsilon k = \Omega - \Omega k$$

BAB 3

IMPLEMENTASI PROGRAM DALAM BAHASA PYTHON

3.1 *Library* Buatan untuk Melengkapi *Library* Bawaan

3.1.1 *File* “InputGambar.py”

Di dalamnya terdapat berbagai fungsi untuk mengambil semua gambar dari set data yang akan digunakan sebagai perbandingan dengan masukan. *Dataset* yang dimaksud sudah disimpan dalam sebuah folder bernama “dataset”, memuat ratusan citra wajah orang yang berbeda-beda.

3.1.2 *File* “faceallig.py”

Face Alignment adalah teknik di mana gambar orang diputar sesuai dengan sudut mata. Teknik ini sebenarnya digunakan sebagai bagian dari proses *pipeline* di mana deteksi wajah dilakukan dengan menggunakan citra. Implementasi perataan wajah ini dapat dengan mudah dilakukan dengan bantuan modul Python cv2 (*computer vision*). Ini memiliki begitu banyak fitur yang disertakan di dalamnya.

Misalkan jika mata orang pada gambar berada pada sudut x ($x! = 180$ derajat) dengan mengacu pada bingkai gambar maka teknik ini akan memutar gambar sesuai dengan sudut dimana sudut x akan sama dengan 180 derajat dengan mengacu pada bingkai gambar.

Sehingga dengan adanya fungsi ini, program dapat mendeteksi dan secara otomatis meng-*crop* gambar wajah seseorang agar bisa diproses dengan baik.

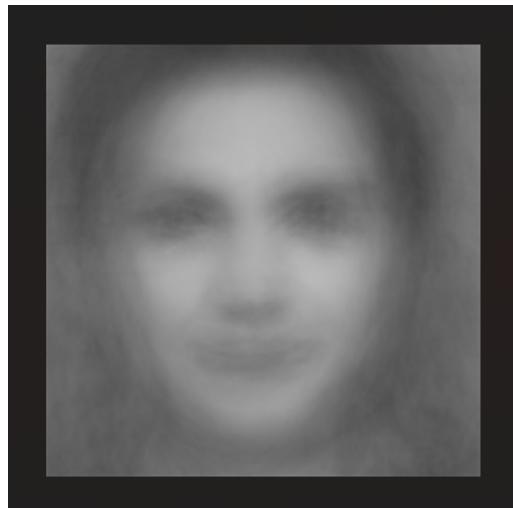
3.1.3 *File* “eigen.py”

Di dalamnya berisi fungsi-fungsi untuk menghitung nilai eigen yang juga digunakan untuk mencari vektor eigen. Disini kami menggunakan metode QR *decomposition* untuk mencari nilai eigen dan vektor eigen.

3.1.4 *File* “Normalize.py”

File ini berisi fungsi *Normalize* yang digunakan untuk menormalisasi suatu matriks yang berupa vektor. Menormalisasi berarti membuat panjang dari vektor tersebut sama dengan 1. Cara yang digunakan adalah dengan menghitung sebuah skalar berupa akar dari jumlah dari kuadrat masing-masing elemen. Kemudian, vektor tersebut dikalikan dengan skalar yang sudah dihitung.

3.2 Mean Face



Gambar 3.2.1

Gambar di atas merupakan hasil operasi *mean* (nilai rata-rata) dari gambar-gambar di *dataset*, yang kemudian diolah dan dikeluarkan lagi dalam bentuk gambar *grayscale*.

3.3 Keberjalan Program Utama

3.3.1 File “main.py”

Kami menempatkan program GUI dan program main menjadi satu kesatuan. Dan saat pengguna melakukan *run* pada program “main.py” maka tampilan UI (*user interface*) akan segera muncul di layar pengguna.

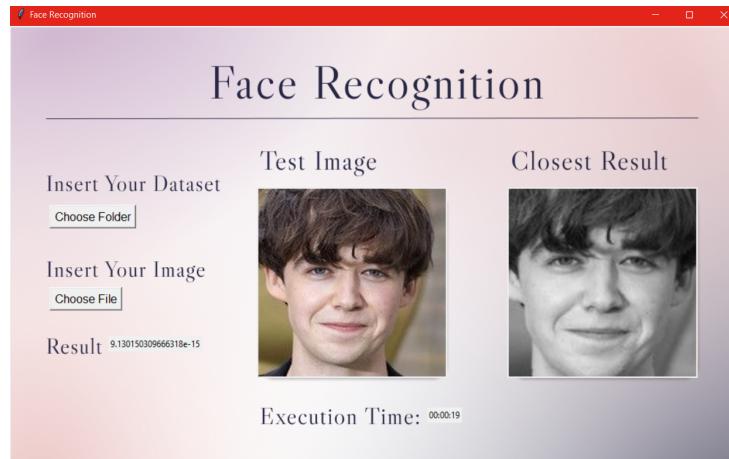
3.4 Pengaplikasian Fitur Bonus

Bonus yang dimaksud berupa penambahan fitur pendekripsi gambar wajah yang diambil secara langsung dari kamera. Set data yang digunakan tetap sama seperti pada fitur sebelumnya. Namun, kamera harus dibuka secara otomatis dalam bentuk aplikasi dengan bantuan *library* OpenCV.

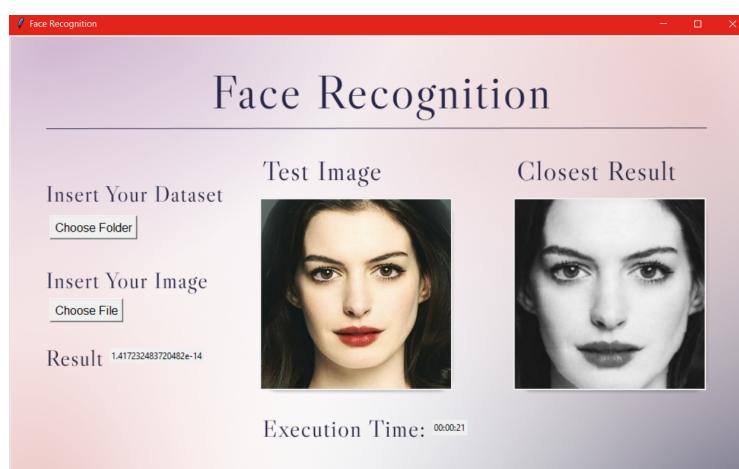
BAB 4

EKSPERIMEN

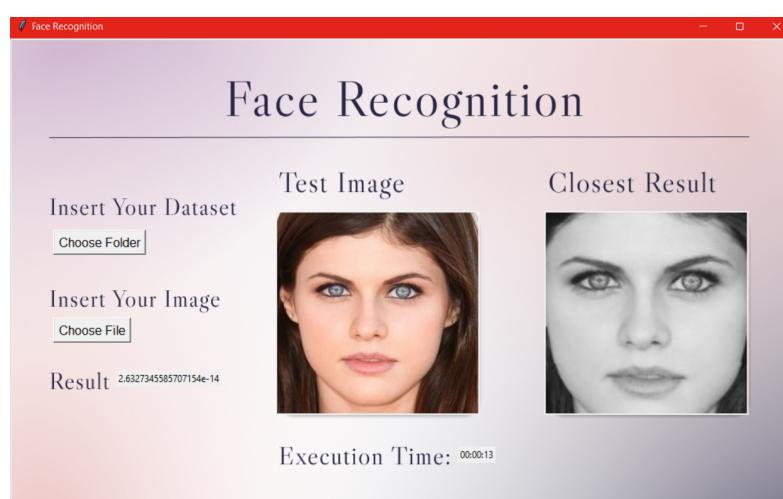
4.1 Mengeluarkan Foto yang Valid



Gambar 4.1.1 Alex Lawter



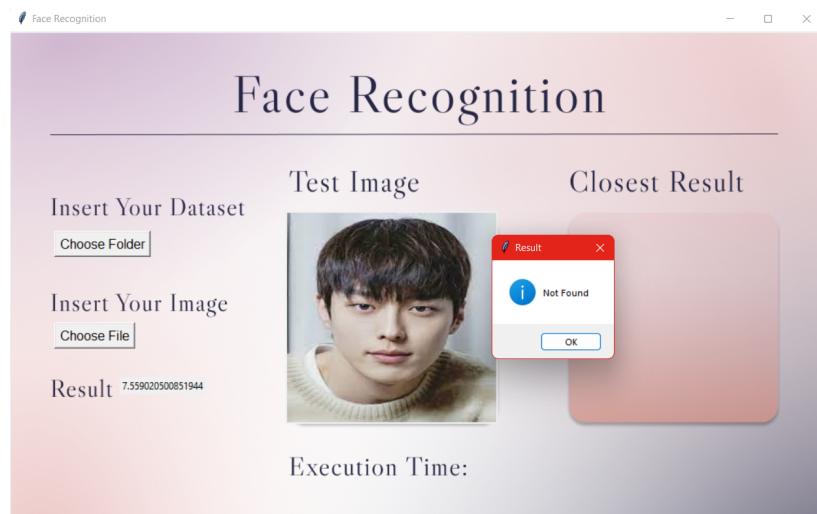
Gambar 4.1.2 Anne Hathway



Gambar 4.1.3 Alexandra Daddario

Percobaan pertama dilakukan dengan memasukkan foto Alex Lawter, Anne Hahtway, dan Alexandra Daddario. Ketiga input tersebut berhasil memberikan keluaran dari set data, berupa foto yang mengandung wajah yang tepat. Dengan waktu eksekusi masing-masing secara berurutan 19 detik, 21 detik, dan 13 detik.

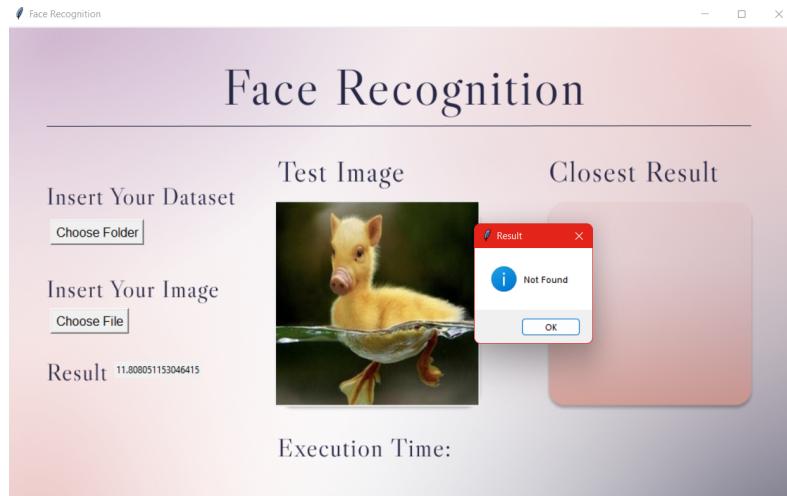
4.2 Keluaran dari Masukan yang Tidak Valid



Gambar 4.2.1 Jan Ki Young



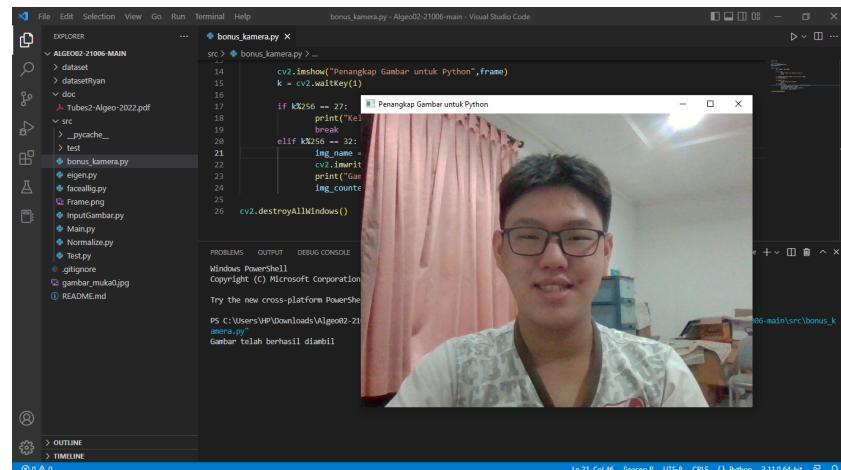
Gambar 4.2.2 IU



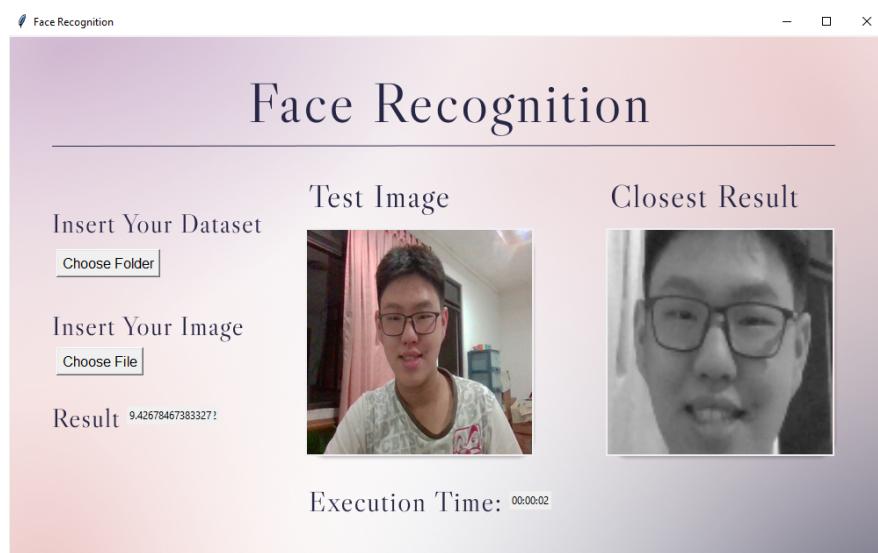
Gambar 4.2.3 BekPig

Percobaan selanjutnya adalah memasukkan foto Jan Ki Young, IU, serta sembarang hewan. Ketiga input tersebut ternyata tidak bisa memberikan keluaran, karena tidak terdapat foto pada *database* yang lumayan mendekati. Selain itu, sudah dijelaskan pula pada **BAB 3** bahwa fungsi di fail “faceallig.py” memastikan program ini hanya bisa mendeteksi wajah manusia, bukan hal lain seperti pada Gambar 4.2.3.

4.3 Inputan Di Kamera (program bonus)



Gambar 4.3.1 Kamera terbuka untuk mengambil gambar secara langsung



Gambar 4.3.2 Keluaran yang dihasilkan apabila menggunakan *database* dan foto masukan diri sendiri

BAB 5

PENUTUP

5.1 Kesimpulan

Di dunia modern, teknologi berkembang dengan sangat pesat. Salah satunya adalah beragam metode pengenalan wajah yang dikomputerisasi. 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 program dapat mempelajari bentuk wajah dan mencocokkan antara gambar pada *database* dengan citra yang akan diidentifikasi.

Secara umum, sistem pengenalan wajah dibagi menjadi 2 jenis, yaitu sistem *feature-based* dan sistem *image-based*. Contoh dari *image-based* adalah algoritma PCA (*Principal Component Analisis*) yang dilibatkan dalam pencarian *eigenface*. Algoritma ini bisa diterapkan dalam bahasa pemrograman Python dengan bantuan berbagai *library*, sampai pengenalan wajah bisa dilakukan berdasarkan *database* masukan sendiri. Selain itu, masukan wajah yang ingin dikenali dapat berupa gambar yang sudah ada, maupun gambar *real time* (diambil langsung dari kamera). Kelancaran program terbukti dengan adanya *test case* pada **BAB 4**.

5.2 Saran

1. Sebaiknya, spesifikasi yang diberikan untuk tugas dengan tingkat kesulitan seperti ini, diberikan secara lebih rinci dan terpadu. Dalam tugas kali ini, langkah demi langkah yang tertera di spesifikasi, beserta tautan-tautan yang dilampirkan, masih belum bisa membawa penulis menyelesaikan program yang diminta.
2. Seharusnya, waktu yang disediakan untuk menyelesaikan tugas ini lebih panjang, sebab banyak sekali eksplorasi mandiri yang harus dilakukan oleh penulis sebelum bisa menuntaskan keseluruhan program.
3. Ada baiknya mengenal terlebih dahulu fungsi-fungsi bawaan dari setiap *library* yang digunakan, sebelum memulai pengimplementasian fitur-fitur.
4. Apabila pada saat mengerjakan, ada anggota kelompok yang merasa penat dan tiba-tiba rusuh, sebaiknya pekerjaan dihentikan terlebih dahulu untuk menyegarkan pikiran. Setelah semuanya kembali segar, program yang dibuat biasanya menjadi lebih bagus dan cepat selesai.

5.3 Refleksi

Tugas besar kali ini memberikan banyak pengalaman dan pengetahuan baru kepada masing-masing anggota kelompok. Keahlian pemrograman dengan bahasa Python juga semakin terasah. Kami juga paham bahwa koordinasi waktu dan kekompakkan tim sangat diperlukan dalam melangsungkan suatu kerja sama.

Meskipun dengan keterbatasan waktu dan pengetahuan, kerja keras serta pengorbanan kami tidaklah sia-sia. Tugas besar kedua untuk mata kuliah ini akhirnya terselesaikan sepenuhnya dengan baik.

LAMPIRAN

Set data yang digunakan:

<https://www.kaggle.com/datasets/herveisburak/pins-face-recognition>

Tautan *repository*:

<https://github.com/Lailabkn/Algeo02-21006.git>

DAFTAR REFERENSI

1. Syarifuddin dkk. 2016. *Aljabar Linear*. Mataram: LPP Mandala.
2. Badan Pengembangan dan Pembinaan Bahasa, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia. 2016. KBBI Daring. *matriks*. (Diakses dari <https://kbbi.kemdikbud.go.id/entri/matriks> tanggal 22 November 2022)
3. Munir, Rinaldi. 2022. *Review Matriks (Bahan Kuliah IF2123 Aljabar Linier dan Geometri #1)*. Program Studi Teknik Informatika, STEI-ITB. (Diakses tanggal 22 November 2022 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-01-Review-Matriks.pdf>)
4. Munir, Rinaldi. 2022. *Nilai Eigen dan Vektor Eigen (Bahan Kuliah IF2123 Aljabar Linier dan Geometri #18)*. Program Studi Teknik Informatika, STEI-ITB. (Diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf> tanggal 22 November 2022)
5. pawangfg. 24 September 2021. *ML | Face Recognition Using Eigenfaces (PCA Algorithm)* (Diakses dari <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/> tanggal 22 November 2022).
6. Alam, Guntur, dkk. 2015. *Implementasi Algoritma Eigenface untuk Face Recognition pada Objek Foto ID Card*. (Diakses dari <http://repository.unib.ac.id/18865/1/4.%20IMPLEMENTASI%20ALGORITMA%20EIGENFACE%20UNTUK%20FACE.pdf> tanggal 22 November 2022)
7. Tanpa nama. *Aplikasi Nilai Eigen dan EigenFace pada Pengenalan Wajah (Face Recognition), Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri, Semester I Tahun 2022/2023.* (Diakses tanggal 22 November 2022 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Tubes-2-Algeo-2022.pdf>)