

IF2123 Aljabar Linear dan Geometri

**IMPLEMENTASI EIGEN FACE DAN NILAI EIGEN
PADA APLIKASI PENGENALAN WAJAH**

Laporan Tugas Besar II

Disusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri
pada Semester 1 (satu) Tahun Akademik 2022/2023



Oleh

Enrique Alifio Ditya 13521142
Vanessa Rebecca Wiyono 13521151
Alisha Listya Wardhani 13521171

Kelompok IN.YOUR.FACE

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG
2022**

DAFTAR PUSTAKA

BAB 1	
DESKRIPSI MASALAH	3
BAB 2	
TEORI SINGKAT	4
BAB 3	
IMPLEMENTASI	7
3.1. Folder Assets	7
3.2. Folder Modules	7
3.3. Folder Test	9
3.4. Index.py	10
3.5. Main.py	10
3.6. Testing.ipynb	10
BAB 4	
EKSPERIMEN	11
4.1. Dataset 1	11
4.1. Dataset 2	13
BAB 5	
KESIMPULAN	15
5.1 Kesimpulan	15
5.2 Saran	15
5.3. Refleksi	15
DAFTAR REFERENSI	17
LAMPIRAN	18

BAB 1 DESKRIPSI MASALAH

Pengenalan wajah merupakan salah satu teknologi biometrik yang bertujuan untuk mengidentifikasi atau mengenali identitas seseorang melalui wajah mereka. Sistem pengenalan wajah ini biasanya bergantung pada database besar yang berisikan kumpulan wajah orang. Selain digunakan untuk membuka kunci gawai (handphone), sistem pengenalan wajah berguna untuk mendeteksi wajah yang berjalan melewati kamera special, sebagai salah satu biometric security. Pada sistem ini, orang yang berada pada daftar watch list dapat diantisipasi sebelumnya.

Pada Tugas Besar II Aljabar Linier dan Geometri ini, dibuat sebuah 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.

Terdapat dua bonus pada Tugas Besar ini, yaitu kamera dan video. Kamera merupakan fitur untuk mengenali wajah secara *realtime*. Teknis kamera dibebaskan dan merupakan fitur tambahan sehingga tidak bisa mensubstitusi fitur upload gambar. Bonus kedua merupakan bonus video pembelajaran implementasi pengenalan wajah. Video ini diunggah ke youtube dengan tujuan untuk mensosialisasikan ilmu mengenai pengenalan wajah yang telah dipelajari.

BAB 2 TEORI SINGKAT

Matriks dan Perkalian Matriks

Matriks merupakan sekumpulan bilangan yang disusun berdasarkan baris dan kolom. Pada pengolahan matriks, terdapat dua jenis perkalian, yaitu perkalian antara matriks dengan matriks dan perkalian antara matriks dengan skalar. Perkalian dua buah matriks, misal matriks A yang berukuran $m \times n$ dengan matriks B yang berukuran $n \times r$ akan menghasilkan matriks C yang berukuran $m \times r$ dengan tiap elemen yang dapat dihitung berdasarkan rumus berikut:

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

(Persamaan 1.1 Perkalian Matriks)

Hal ini berbeda dengan perkalian matriks dengan skalar, dimana hasil perkalian didapat dari perkalian setiap elemen pada matriks dengan skalar.

Nilai Eigen dan Vektor Eigen

Misal A merupakan matriks berukuran $n \times n$, maka vektor tidak nol di R^n disebut sebagai vektor eigen dari A jika Ax sama dengan perkalian skalar λ dengan x , yaitu $Ax = \lambda \times x$ disebut sebagai nilai eigen dari A dan x disebut sebagai vektor eigen dari A. Nilai eigen menyatakan nilai karakteristik dari sebuah matriks berukuran $n \times n$, dan vektor eigen menyatakan vektor kolom yang menghasilkan vektor lain kelipatan vektor itu sendiri apabila dikalikan dengan sebuah matriks $n \times n$.

Eigenface

Eigenface merupakan suatu metode pengenalan wajah dengan menggunakan nilai eigen dan vektor eigen yang diturunkan dari matriks kovarian sejumlah citra wajah (*training image*). Dengan Eigenface, gambar-gambar yang digunakan untuk membangun matriks kovarian akan dibentuk menjadi himpunan basis sehingga menghasilkan pengurangan dimensi dengan memungkinkan kumpulan gambar dasar yang lebih kecil untuk mewakili gambar pelatihan asli. Kemudian, akan dilakukan perbandingan representasi wajah oleh himpunan basis. Metode ini dikembangkan oleh Sirovich dan Kirby dan digunakan oleh Matthew Turk dan Alex Pentland untuk mengidentifikasi wajah.

Berikut langkah-langkah perhitungan algoritma Eigenface:

Algoritma Eigenface

1. Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh training image, ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$)

$$S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M) \quad (1)$$

2. Langkah kedua adalah ambil nilai rata-rata atau mean (Ψ)

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

3. Langkah ketiga kemudian cari selisih (Φ) antara nilai training image (Γ_i) dengan nilai tengah (Ψ)

$$\phi_i = \Gamma_i - \Psi \quad (3)$$

4. Langkah keempat adalah menghitung nilai matriks kovarian (C)

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (4)$$

$$L = A^T A \quad L = \phi_m^T \phi_n$$

5. Langkah kelima menghitung eigenvalue (λ) dan eigenvector (v) dari matriks kovarian (C)

$$C \times v_i = \lambda_i \times v_i \quad (5)$$

6. Langkah keenam, setelah eigenvector (v) diperoleh, maka eigenface (μ) dapat dicari dengan:

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k \quad (6)$$

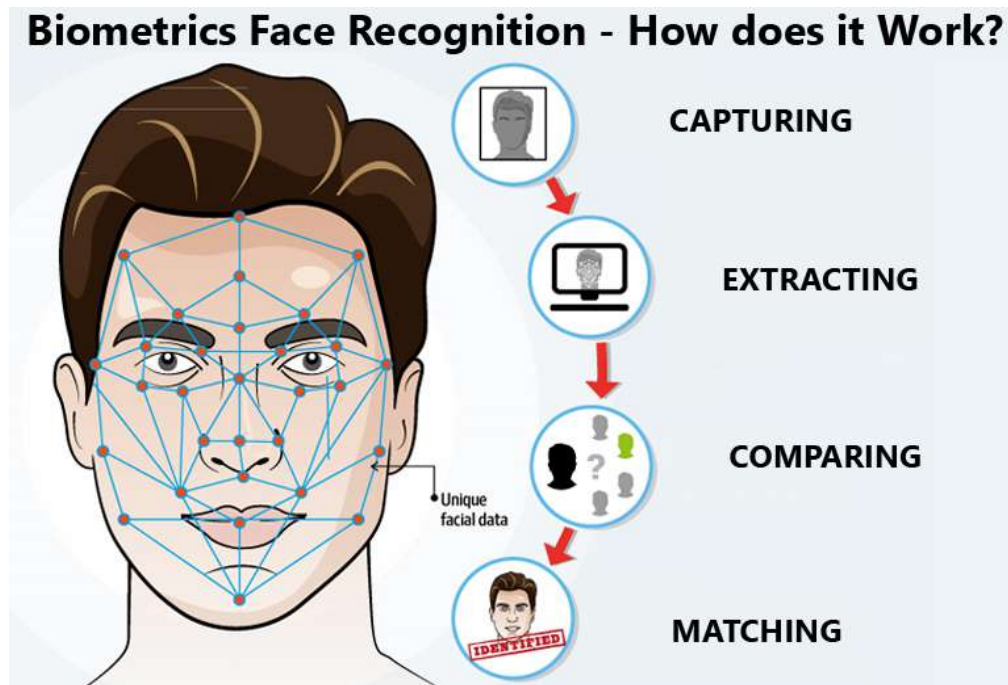
$$l = 1, \dots, M$$

Sumber: spesifikasi tugas besar 2 Algeo

Face recognition

Face recognition merupakan salah satu implementasi nyata dari metode eigenface, dimana face recognition sendiri merupakan teknologi biometrik yang digunakan untuk mengidentifikasi wajah seseorang dengan berbagai tujuan kegunaan terlebih dalam tujuan keamanan. Setiap wajah manusia dalam kelompok ras merupakan kombinasi dari beberapa lusin bentuk primer, sehingga bentuk kepala rata-rata (*meanhead*) dapat dianalisis dan digunakan dalam metode eigenface dengan euclidean distance sebagai classifier yang mampu menghasilkan tingkat akurasi cukup tinggi. *Euclidean distance* didapat dari *training weight* yang merupakan hasil perkalian antara

Eigen face dengan fitur prominent pada wajah, dan *test weight* yang merupakan hasil perkalian antara Eigenface transpose dengan hasil pengurangan dari matrix training set dengan average set dari suatu wajah. Secara sederhana, matriks fitur menonjol dari setiap orang akan dikurangi dengan matriks muka rata-rata banyak orang.



Source:

<https://www.starlinkindia.com/blog/wp-content/uploads/2019/05/Biometrics-Face-Recognition-How-Does-it-work.jpg>

BAB 3

IMPLEMENTASI

Dalam Tugas Besar II, terdapat 2 folder utama: src dan doc. Folder src membuat source code pada program. Pada folder src, terdapat folder data (assets, samples, test_sets, dan training_sets) serta folder modules yang berisi fungsi yang menyokong file main.py. Adapun program utama berupa GUI terdapat pada main.py. Berikut merupakan rincian, deskripsi, serta implementasi dari setiap file tersebut.

3.1. Folder Assets

Folder assets berisikan aset-aset berupa gambar yang akan digunakan pada GUI (Graphic User Interface) program. Pada folder ini, terdapat 18 gambar dan satu font type. Asset ini kemudian akan direferensikan pada file main.py sebagai program utama.

3.2. Folder Modules

Folder modules berisi modul-modul yang akan digunakan pada program. Pada folder modules terdapat 5 file: config.py, eigen.py, face_recognition.py, util.py, dan webcam.py. Berikut merupakan penjelasan dari setiap file pada module.

3.2.1. Config.py

File ini berisi konfigurasi path penempatan program utama.

<pre>import os</pre>	Mengimpor OS Module yang menyediakan fungsi untuk membuat dan menghapus direktori (folder), mengambil isinya, mengubah dan mengidentifikasi direktori saat ini.
<pre>ROOT_DIR = os.path.realpath(os.path.join(os.path.dirname(__file__), '..'))</pre>	Inisialisasi ROOT_DIR yang merupakan path penempatan folder src.

3.2.2. Eigen.py

File ini berisi fungsi terkait pencarian nilai eigen dan vektor eigen.

<pre>function qrDecomp(input mCov : <u>matriks</u>) { Menghasilkan QR Decomposition dari masukan berupa Covariance Matrix }</pre>	Fungsi yang menghasilkan dekomposisi qr dengan masukan berupa matrix kovarian.
<pre>function eig(input mCov : <u>matriks</u>) { Menghasilkan nilai eigen dan vektor eigen dari masukan berupa Covariance Matrix }</pre>	Fungsi yang menghasilkan nilai eigen dan vektor eigen dengan masukan berupa matriks kovarian. Fungsi ini memanggil fungsi qrDecomp dan mengembalikan tuple yang berisi nilai eigen dan vektor eigen.

3.2.3. Face_recognition.py

File ini berisi fungsi yang berkaitan dengan pemrosesan face recognition pada program

<pre>import numpy as np</pre>	Mengimpor Numerical Python Module yang
--------------------------------------	--

	merupakan library untuk memudahkan bekerja dengan array.
<code><u>import</u> cv2</code>	Mengimpor OpenCV module yang yang dirancang untuk memecahkan masalah berkaitan dengan <i>computer vision</i> .
<code><u>import</u> os</code>	Mengimpor OS Module yang menyediakan fungsi untuk membuat dan menghapus direktori (folder), mengambil isinya, mengubah dan mengidentifikasi direktori saat ini.
<code><u>import</u> modules.eigen as eg <u>from</u> modules.config import ROOT_DIR</code>	Mengimpor fungsi yang telah dibuat pada file lain.
<code><u>function</u> average_face(input M: <u>matriks</u>) { Mengembalikan vektor rata-rata wajah }</code>	Fungsi yang mengembalikan vektor rata-rata wajah dari dataset dengan masukkan berupa matrix
<code><u>function</u> deviation(input M: <u>matriks</u>) { Mengembalikan vektor hasil pengurangan antara rata-rata wajah dengan matriks }</code>	Fungsi yang mengembalikan pengurangan antara vektor wajah rata-rata dengan masukkan yang berupa matrix.
<code><u>function</u> covariance(input M: <u>matriks</u>) { Mengembalikan matriks kovarian }</code>	Fungsi yang mengembalikan matriks kovarian.
<code><u>procedure</u> eigenface(input M: <u>matriks</u>, rank: <u>integer</u>) { I.S. Matriks sembarang tidak kosong } { F.S. Eigenvektor terurutkan dan ternormalisasi. EigenFace terdefinisi }</code>	Prosedur yang menormalisasi dan mengurutkan eigenvector dan mengembalikan eigenface dengan masukan berupa matriks dan rank
<code><u>function</u> euc_distance(input v1, v2: <u>matriks</u>) { Mengembalikan euclidean_distance dari masukan dua buah vektor }</code>	Fungsi yang mengembalikan euclidean distance dengan memasukkan dua buah vektor.

3.2.4. Utils.py

File ini berisi fungsi utilitas yang berfungsi sebagai fungsi bantuan dalam program utama.

<code><u>import</u> numpy as np</code>	Mengimpor Numerical Python Module yang merupakan library untuk memudahkan bekerja dengan array.
<code><u>import</u> cv2</code>	Mengimpor OpenCV module yang yang dirancang untuk memecahkan masalah berkaitan dengan <i>computer vision</i> .
<code><u>import</u> os</code>	Mengimpor OS Module yang menyediakan fungsi untuk membuat dan menghapus direktori (folder), mengambil isinya, mengubah dan mengidentifikasi direktori saat ini.
<code><u>import</u> shutil</code>	Mengimport Shutil Module yang menyediakan banyak fungsi operasi tingkat tinggi pada file dan

	kumpulan file.
<code>from zipfile import ZipFile</code>	Mengimpor modul zipfile yang berfungsi untuk mengolah file berupa zip.
<code>import matplotlib.pyplot as plt</code>	Mengimport modul Matplotlib yang pada program ini memiliki fungsi utama untuk menunjukkan gambar dari matrix.
<code>import modules.eigen as eg</code> <code>from modules.config import ROOT_DIR</code>	Mengimpor fungsi yang telah dibuat pada file lain.
<code>procedure show_image(input imgFile: string)</code> { I.S. imgFile terdefinisi dan merupakan sebuah pathfile pada direktori } { F.S. Tampilkan gambar imgFile pada GUI }	Prosedur yang menampilkan gambar pada GUI dengan masukan berupa suatu file image.
<code>procedure show_image_plt(input M: matriks)</code> { I.S. M sembarang, terdefinisi } { F.S. Tampilkan gambar dari matriks }	Prodesur yang mengembalikan display image dari input berupa matriks.
<code>function norm(input img: matriks)</code> { Mengembalikan array yang merupakan normalisasi daripada matriks img masukan }	Fungsi yang mengembalikan normalisasi greyscale terhadap matriks image.
<code>procedure extract(input filename: string)</code> { I.S. filename terdefinisi dan merupakan sebuah pathfile pada direktori } { F.S. ZIP file terekstrak }	Prosedur yang digunakan untuk mengekstrak file berupa zip.
<code>function list_files(input filepath: string, filetype: string)</code> { Mengembalikan array yang berisikan pathfiles dari sebuah direktori }	Fungsi yang mengembalikan sebuah array berisikan pathfiles dari sebuah direktori.
<code>function image_to_matrix(input dirname: string)</code> { Mengembalikan kumpulan matriks dari folder dataset yang tersedia }	Fungsi yang mengembalikan kumpulan matriks dari folder dataset yang tersedia.

3.2.6. Webcam.py

File ini berisikan fungsi yang menjalankan webcam pada GUI.

<code>procedure webcamFunc()</code> { Prosedur yang membuka kamera pada GUI dan mengambil foto dengan memencet tombol c pada keyboard }	Prosedur yang membuka kamera dan mengambil foto secara realtime lalu mengatur ukuran foto tersebut menjadi persegi dengan rasio 1:1 dan hasilnya akan disimpan sebagai "hasilWebcam.jpg".
--	---

3.3. Folder Test

Folder ini berisikan dataset yang digunakan dalam uji citra pada program. Folder ini berisikan 3 subfolder: sample, test_sets, dan ungrouped.

3.3.1. Folder sample

Folder test set berisi gambar-gambar yang akan digunakan untuk keperluan training program.

Pada folder sample, terdapat 18 orang yang berbeda. Setiap orang memiliki 4-10 gambar yang dijadikan data acuan pada program ini. Pada sample yang terdapat di Tubes II ini, kami mengambil dataset pribadi berupa muka-muka mahasiswa STEI 2021.

3.3.2. Folder Test_Sets

Folder ini berisi gambar yang akan menjadi uji citra pada sistem pengenalan wajah. Folder ini terdiri dari 9 gambar, dengan rincian 5 orang yang berbeda.

3.3.2. Folder Ungrouped

Hampir sama seperti folder sample, folder ini berisi gambar yang akan digunakan untuk keperluan training program. Bedanya, folder ini tidak terklasifikasi menjadi sub-sub folder. Folder ini berisi gambar dengan rincian satu gambar per orang. Gambar pada dataset ini merupakan gambar hasil pribadi yang kami cari sendiri.

3.4. Index.py

File ini merupakan file yang menyimpan fungsi yang berperan sebagai algoritma utama dari face recognition, dimana kompilasi atas seluruh proses face recognition dilakukan.

<pre>procedure index(input dataset : string, isDatasetZip : boolean) { I.S. dataset terdefinisi yang merupakan pathfile pada direktori program} { F.S. training_set, training_weight dan eigen_face terdefinisi }</pre>	<p>Prosedur yang mengembalikan training_set, training_weight dan eigen_face berdasarkan direktori dataset.</p>
<pre>procedure recognize(input test: <u>string</u>, training_weight : <u>matriks</u>, training_set : matriks, eigen_face : <u>matriks</u>) { Mengembalikan hasil dari sistem pengenalan wajah}</pre>	<p>Prosedur yang mengembalikan hasil dari sistem pengenalan wajah.</p>

3.5. Main.py

Main.py merupakan file utama yang dipanggil untuk menjalankan program karena penggunaannya merupakan implementasi dari fungsi-fungsi lainnya. Main.py juga mencakup konfigurasi atas GUI yang telah disambungkan dengan algoritma untuk face recognition.

3.6. Testing.ipynb

Testing.ipynb merupakan file yang digunakan oleh penulis untuk mendebug dan melihat alur program secara modular. Hal ini digunakan oleh penulis untuk melihat secara langsung program yang digunakan pada main.py.

BAB 4 EKSPERIMEN

Berikut merupakan eksperimen dari Tugas Besar II Aljabar Linier dan Geometri.



4.1. Dataset 1

Pada percobaan pertama, digunakan dataset sebanyak 120 citra. Dataset tersebut merupakan kumpulan wajah mahasiswa STEI ITB yang didapat dengan dikumpulkan secara pribadi. Pada tahap ini, diperoleh durasi training adalah 15.002 sekon. Setelah melalui training, didapat wajah rata-rata dan eigenfaces berikut:


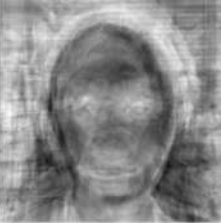








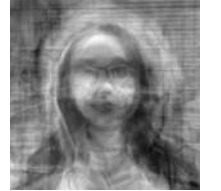






Gambar 4.1. Sampel dari Dataset 1

Tabel 4.1. Average Face dan EigenFace dari Dataset 1

Average Face	Contoh EigenFaces #1
<p style="text-align: center;">Average Face</p> 	<p style="text-align: center;">Eigenface #1</p> 

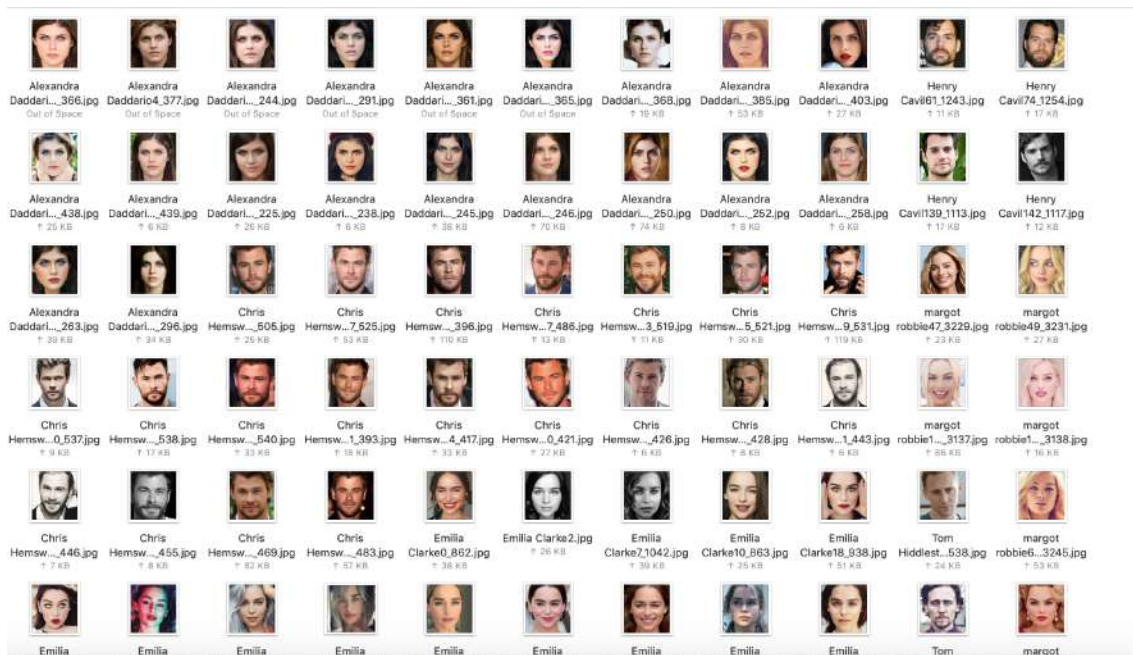
Tabel 4.2. Unit Testing dari Dataset 1

No.	Citra Uji	Reconstruction Image	Citra Identified Person	Euclidean Distance
1.				Euclidean distance: 57.96191389748725
Analisis: Pengidentifikasian citra uji berhasil dengan jarak euclidean yang cukup besar yaitu sekitar 57.96 sehingga gambar rekonstruksi cukup samar				
2.				Euclidean distance: 25.01912039987492
Analisis: Pengidentifikasian citra uji berhasil dengan jarak euclidean yang relatif lebih kecil ketimbang hasil uji lainnya yaitu sekitar 25.01 sehingga gambar rekonstruksi tampak mirip dengan identified person walau masih samar				
3.				Euclidean distance: 55.419193510835925
Analisis: Pengidentifikasian citra uji berhasil dengan jarak euclidean yang cukup besar yaitu sekitar 55.41 sehingga gambar rekonstruksi cukup samar				
4.				Euclidean distance: 38.70445294932391
Analisis: Pengidentifikasian citra uji berhasil dengan jarak euclidean sekitar 25.01 sehingga gambar rekonstruksi cukup samar namun masih mirip dengan identified person				

5.				Euclidian distance: 31.40143615057383
Analisis: Pengidentifikasian citra uji berhasil dengan jarak euclidian sekitar 31.40 sehingga gambar rekonstruksi cukup samar namun masih mirip dengan identified person				



4.1. Dataset 2

Pada percobaan selanjutnya, kami menggunakan dataset sebanyak 80 gambar. Dataset tersebut merupakan kumpulan wajah selebriti yang didapat dari sumber <https://www.kaggle.com/datasets/vasukipatel/face-recognition-dataset>. Pada tahap ini, diperoleh durasi training adalah 23 detik. Setelah melalui training, didapat wajah rata-rata dan eigenfaces berikut:












Gambar 4.2. Sampel dari Dataset 2

Tabel 4.1. Average Face dan EigenFace dari Dataset 2

Average Face	Contoh EigenFaces #1
<p>Average Face</p> 	<p>Eigenface #1</p> 

Tabel 4.2. Unit Testing dari Dataset 2

No.	Citra Uji	Reconstruction Image	Citra Identified Person	Euclidean Distance
1.				Euclidean Distance: 42.03608451
Analisis: Pengidentifikasian berhasil dengan jarak euclidean yang relatif dekat serta image rekonstruksi yang cukup vivid.				
2.				Euclidean Distance: 66.707795407
Analisis: Pengidentifikasian berhasil dengan jarak euclidean yang cukup besar serta rekonstruksi sehingga rekonstruksi samar.				
3.				Euclidean Distance: 45.68441939
Analisis: Pengidentifikasian citra uji gagal dengan jarak euclidean 45.68 dan dihasilkan citra uji yang mirip namun samar. Hal ini mungkin disebabkan oleh struktur muka dan fitur wajah seperti rambut wajah dan garis rahang yang mirip antar wajah uji dengan wajah hasil identifikasi.				

BAB 5

KESIMPULAN

5.1 Kesimpulan

Pada Tugas Besar II IF2123 Aljabar Linier dan Geometri ini telah diimplementasikan sistem pengenalan wajah menggunakan metode Eigenface. Pada tugas besar ini, pertama-tama digunakan sebuah folder dataset yang nantinya diubah menjadi sekumpulaj matriks berisi wajah. Melalui average face dan eigen face, dapat dicari euclidean distance antara citra wajah uji dengan wajah yang ada pada dataset. Wajah dengan euclidean distance minimum merupakan wajah yang diidentifikasi paling mirip dengan wajah uji.

Dengan demikian, penulis menyimpulkan bahwa melalui Tugas Besar II IF2123 Aljabar Linier dan Geometri, dapat dibuat sebuah algoritma sistem pengenalan wajah melalui metode EigenFace.

5.2 Saran

Tugas Besar II IF2123 Aljabar Linier dan Geometri Semester II Tahun 2022/2023 menjadi salah satu tugas yang memberikan banyak pelajaran bagi penulis. Berdasarkan pengalaman penulis mengerjakan tugas ini, berikut merupakan saran untuk pembaca yang ingin melakukan atau mengerjakan hal serupa.

1. Spesifikasi program meminta penulis untuk mengimplementasikan GUI (Graphical User Interface) yang belum diajarkan oleh kuliah IF2123 Semester 1 Tahun 2022/2023. Penulis merekomendasikan untuk menyediakan waktu untuk eksplor dan mempelajari tentang hal-hal diluar materi kuliah dan tidak dibatasi oleh materi yang diajarkan di kelas.
2. Keefektifan dalam kerja sama tim adalah hal yang penting dalam mengerjakan tugas ini. Pembagian tugas yang jelas menjadi salah satu faktor terselesaikannya tugas ini dengan cepat. Penggunaan alat seperti Google Documents dan Github Version Control juga sangat disarankan untuk mempercepat workflow.
3. Mengingat bahwa program mengolah dataset yang besar dan banyak, diperlukan optimisasi pada program dan gunakan library yang sudah tersedia (jika diperbolehkan). Library yang sudah tersedia ini sudah mengoptimisasi algoritma yang dijalankan sehingga dapat mempercepat proses program.

5.3. Refleksi

Tugas Besar II IF2123 Aljabar Linier dan Geometri Semester II Tahun 2022/2023 merupakan salah satu tugas besar yang menantang dan menarik untuk dikerjakan pada semester ini. Proses pengerjaan tugas ini tentunya melalui berbagai rintangan. Salah satunya adalah mencari sumber yang valid terkait penyelesaian algoritma yang berkaitan dengan pencarian

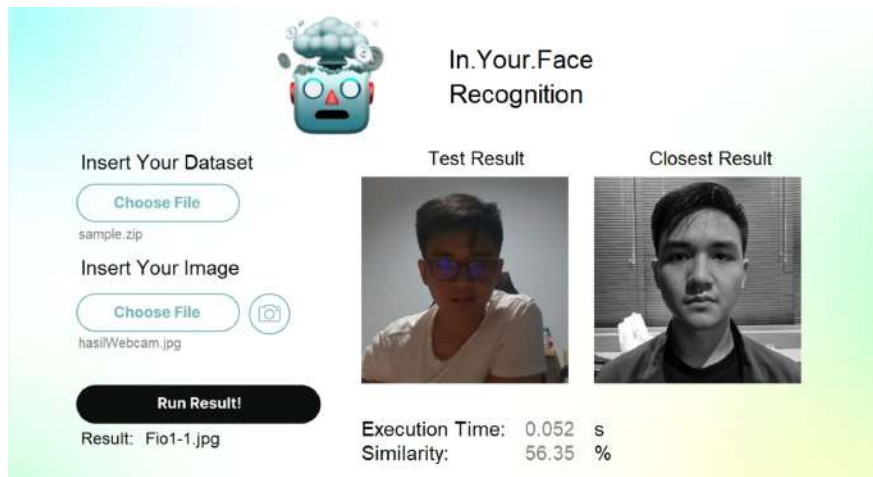
Eigen Faces. Terlepas dari rintangan tersebut, penulis merasakan apresiasi yang besar terhadap sistem pengenalan wajah serta implementasinya, baik secara matematis ataupun dalam bidang teknologi. Kegunaan sistem pengenalan wajah juga dapat dijumpai pada sebagian besar kehidupan kita di jaman sekarang, contohnya pengenalan wajah pada tiket masuk KAI. Pada tugas ini, penulis diberi kesempatan untuk mengimplementasikan pengolahan citra melalui eigen faces dan timbul rasa apresiasi yang tinggi dan kebanggaan dalam keberhasilan untuk memproduksi sebuah teknologi dari sekumpulan gambar. Dalam mengerjakan tugas ini, penulis juga merasakan senangnya berhasil untuk melihat sebuah program asli (melalui gui) dan mengimplementasikannya dalam kurun waktu di bawah tiga minggu.

DAFTAR REFERENSI

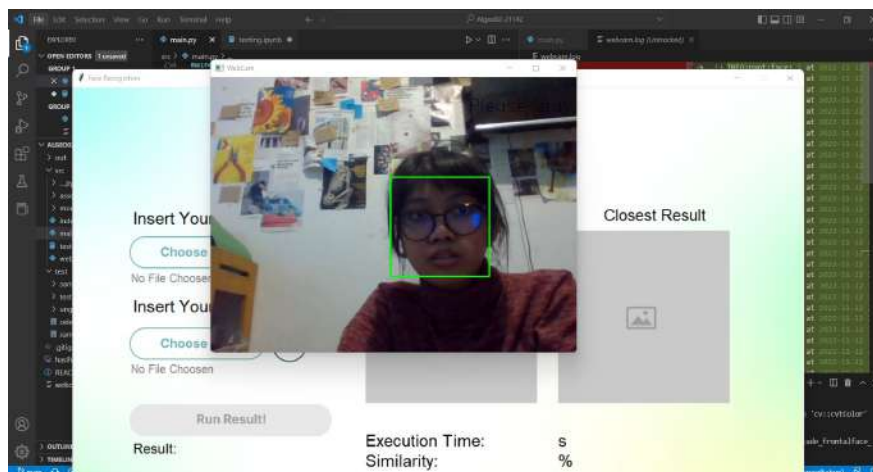
- Spesifikasi Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Sistem Persamaan Linier, Determinan, dan Aplikasinya Semester I Tahun 2022/2023. (2022). Program Studi Teknik Elektro dan Informatika Institut Teknologi Bandung.
- Mishra, M., Srivastava, U., Sharma, A., & Srivastava, U. (2021, July 14). Face Detection Using a real-time webcam and Implemented with Haar cascade Algorithm. datamahadev.com. <https://datamahadev.com/face-detection-using-a-real-time-webcam-and-implemented-with-haar-cascade-algorithm/>
- How Does A Computer Calculate Eigenvalues? (2017, October 4). <https://madrury.github.io/jekyll/update/statistics/2017/10/04/qr-algorithm.html>
- GeeksforGeeks. (2021, September 24). ML | Face Recognition Using Eigenfaces (PCA Algorithm). <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>
- Munir, R. (2022). Algeo #21 Nilai Eigen dan Vektor Eigen. Retrieved from Homepage Rinaldi Munir: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf>
- Munir, R. (2022). Algeo #22 Nilai Eigen dan Vektor Eigen. Retrieved from Homepage Rinaldi Munir: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian2-2022.pdf>
- Munir, R. (2022). Algeo #23 Singular Value Decomposition. Retrieved from Homepage Rinaldi Munir: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Algeo-20-Singular-value-decomposition-Bagian1-2022.pdf>
- Munir, R. (2022). Algeo #24 Singular Value Decomposition. Retrieved from Homepage Rinaldi Munir: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Algeo-21-Singular-value-decomposition-Bagian2-2022.pdf>
- Munir, R. (2022). Algeo #25 Dekomposisi LU. Retrieved from Homepage Rinaldi Munir: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Algeo-22-Dekomposisi-LU.pdf>

LAMPIRAN

Screen Shot GUI



Screen Shot Kamera



Repository Github

Berikut merupakan link repository Github untuk program penulis.

<https://github.com/AlifioDitya/Algeo02-21142>

Tautan Youtube

Berikut merupakan link youtube untuk bonus video

<https://youtu.be/YmWUue5Nbsk>