

LAPORAN
TUGAS BESAR 2
ALJABAR LINIER DAN GEOMETRI
APLIKASI NILAI EIGEN DAN *EIGENFACE* PADA
PENGENALAN WAJAH



Disusun Oleh:

13521020	Varraz Hazzandra Abrar
13521053	Athif Nirwasito
13521085	Addin Munawwar Yusuf

DAFTAR ISI

BAB 1 DESKRIPSI MASALAH

BAB 2 TEORI SINGKAT

BAB 3 IMPLEMENTASI PROGRAM

BAB 4 EKSPERIMEN

BAB 5 KESIMPULAN, SARAN, DAN REFLEKSI

REFERENSI

LAMPIRAN

DAFTAR GAMBAR

Gambar 1.1 Proses Pengenalan Wajah

Gambar 2.1.1 Perkalian Matriks

Gambar 2.2.1 Gambaran Vektor Eigen

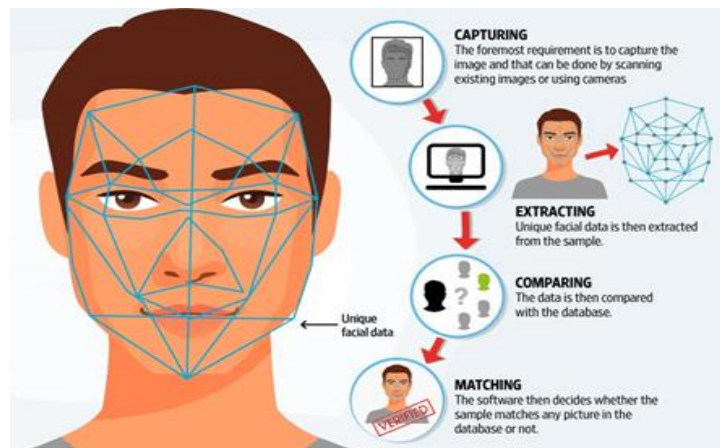
Gambar 3.1 Hasil Tes Face Recognition Dengan Sampel Citra Wajah Keanu Reeves

Gambar 3.2 Hasil Rekonstruksi Gambar dengan Eigenface

Gambar 3.3 Hasil Input Melalui Kamera

BAB 1

DESKRIPSI MASALAH



Gambar 1.1 Proses Pengenalan Wajah

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.

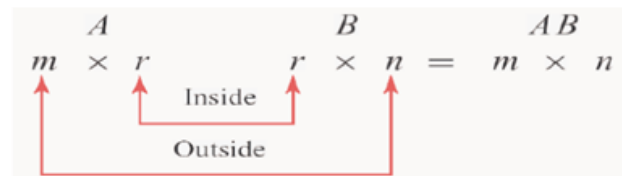
Terdapat berbagai teknik untuk memeriksa citra wajah dari kumpulan citra yang sudah diketahui seperti jarak Euclidean dan *cosine similarity*, principal component analysis (PCA), serta Eigenface. Pada Tugas ini, akan dibuat sebuah program pengenalan wajah menggunakan Eigenface.

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.

BAB 2

TEORI SINGKAT

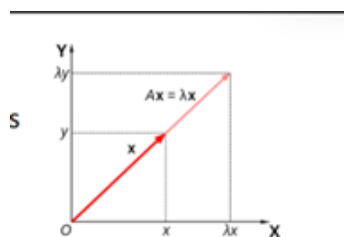
2.1 PERKALIAN MATRIKS



Gambar 2.1.1 Perkalian Matriks

Matriks $A_{m \times r}$ dikalikan dengan matriks $B_{r \times n}$ dan menghasilkan matriks $C_{m \times n}$. Misalkan $A = [a_{ij}]$ dan $B = [b_{ij}]$, maka $C = A \cdot B = [c_{ij}]$, $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$. Hal ini berlaku dengan syarat, yaitu jumlah kolom A sama dengan jumlah baris B .

2.2 VEKTOR EIGEN



Gambar 2.2.1 Gambaran Vektor Eigen

Vektor eigen x adalah matriks kolom yang jika dikalikan dengan sebuah matriks $n \times n$, hasilnya adalah vektor lain yang merupakan kelipatan vektor itu sendiri.

2.3 NILAI EIGEN

Jika A adalah matriks $n \times n$, vektor tidak-nol x di R_n disebut vektor eigen dari A jika Ax sama dengan perkalian suatu skalar λ dengan x ($Ax = \lambda x$). Skalar λ disebut nilai eigen dari A , dan x dinamakan vektor eigen yang berkoresponden dengan λ . Kata “eigen” berasal dari Bahasa Jerman yang berarti “asli” atau “karakteristik”. Hal ini bermakna bahwa nilai eigen menyatakan nilai karakteristik dari sebuah matriks yang berukuran $n \times n$.

2.4 EIGENFACE

Eigenface adalah metode pengenalan wajah (face recognition) berbasis vektor eigen dan nilai eigen yang berguna dalam persoalan-persoalan di dalam *computer vision*. Eigenface dikembangkan oleh Sirovich and Kirby. Lalu, Matthew Turk dan Alex Pentland menggunakan pengembangan tersebut untuk klasifikasi wajah.

Vektor eigen berasal dari matriks kovarian dari sejumlah citra wajah latih (*training image*). Eigenface membentuk himpunan basis dari semua gambar yang digunakan untuk membangun matriks kovarian. Ini menghasilkan pengurangan dimensi dengan memungkinkan kumpulan gambar dasar yang lebih kecil untuk mewakili gambar pelatihan asli. Klasifikasi dapat dicapai dengan membandingkan bagaimana wajah direpresentasikan oleh himpunan basis .

BAB 3

IMPLEMENTASI PROGRAM

Program pengenalan wajah dengan eigenface ini dibuat dengan menggunakan bahasa pemrograman python. Dalam pengaplikasiannya, digunakan beberapa library python untuk mempermudah pekerjaan. Library-library tersebut adalah:

1. tkinter v8.6 : Digunakan dalam pembuatan Graphical User Interfaces (GUI)
2. numpy v1.23.4 : Digunakan dalam melakukan operasi array dan matrix sederhana

3. Pillow v9.3.0 : Digunakan dalam Image Processing (resize, conversion, GUI)
4. OpenCV v4.6.0 : Digunakan dalam Image Processing, Conversion, dan input kamera
5. matplotlib v3.6 : Digunakan dalam melakukan testing algoritma
6. python time : Digunakan untuk tracking waktu eksekusi
7. python os : Digunakan dalam pembacaan dan penulisan file ataupun folder
8. python math : Digunakan untuk operasi matematika sederhana (square root)

Dalam pengaplikasiannya, algoritma disebar ke dalam 7 modul. Rincian dari masing-masing modul adalah sebagai berikut:

A. main.py

main.py berfungsi sebagai entry point dari program. main.py hanya berisi satu instruksi, yaitu menjalankan prosedur `START_APP()` yang akan menjadi trigger untuk aplikasi dimulai

B. app.py

app.py berperan sebagai penghubung utama antar modul di dalam program. app.py berhubungan langsung dengan user dalam menerima input dan mengeluarkan output sehingga GUI aplikasi juga diimplementasikan di sini. Input-input dari user tersebutlah yang akan menjadi *trigger* bagi app.py untuk menjalankan fungsi-fungsi dari modul lain.

C. face_recog.py

face_recog.py adalah modul utama yang menjalankan algoritma dari face recognition itu sendiri. modul ini memiliki fungsi yang menjalankan proses face recognition dari awal hingga akhir, mulai dari mengekstrak dataset, mencari bobot dataset dan bobot gambar tes, hingga pencarian gambar terdekat.

D. eigen.py

eigen.py adalah modul yang berfungsi dalam pencarian nilai-nilai eigen dan eigen vector. Modul ini sangat berperan dalam proses face recognition untuk mencari nilai eigen face, hingga bobot dari image terhadap eigenfaces.

E. euclidean_distance.py

euclidean_distance.py berperan dalam melakukan perhitungan jarak antar vektor. Dalam proses face recognition, modul ini digunakan untuk mencari gambar yang terdekat antara gambar tes dan dataset berdasarkan bobotnya terhadap eigenfaces.

F. image_handler.py

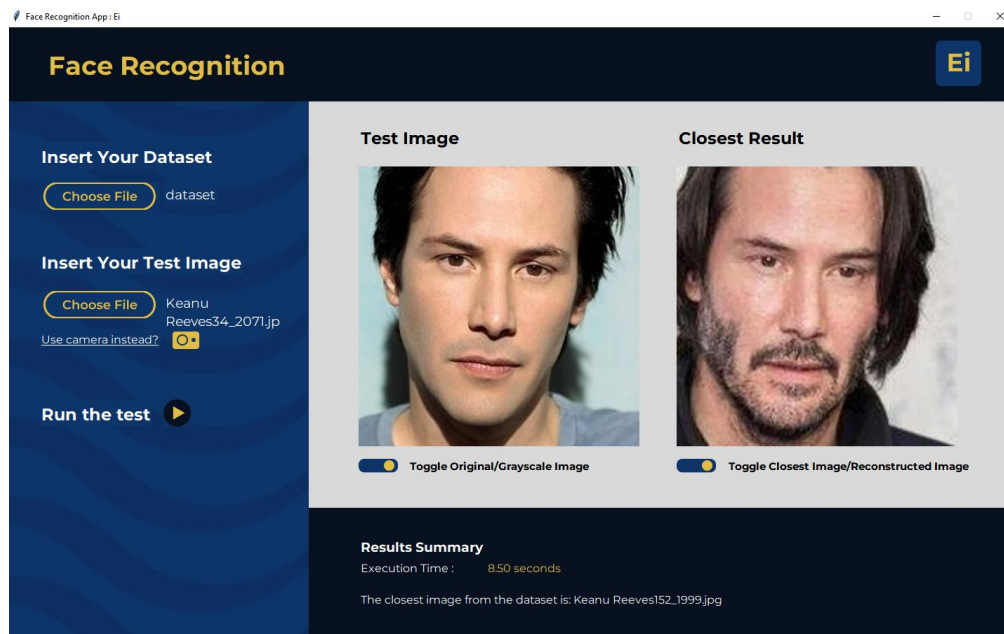
image_handler.py digunakan untuk pembacaan gambar, konversi gambar, hingga pemrosesan ukuran gambar, baik itu untuk keperluan GUI maupun algoritma.

G. camera_input.py

camera_input.py adalah modul yang berperan dalam menerima input gambar user melalui kamera. Input tersebut dapat disesuaikan, untuk keperluan training maupun testing.

Berikut ini adalah hasil screenshot hasil implementasi dari aplikasi yang dibuat.

1. Hasil Test Face Recognition terhadap Citra Wajah Keanu Reeves



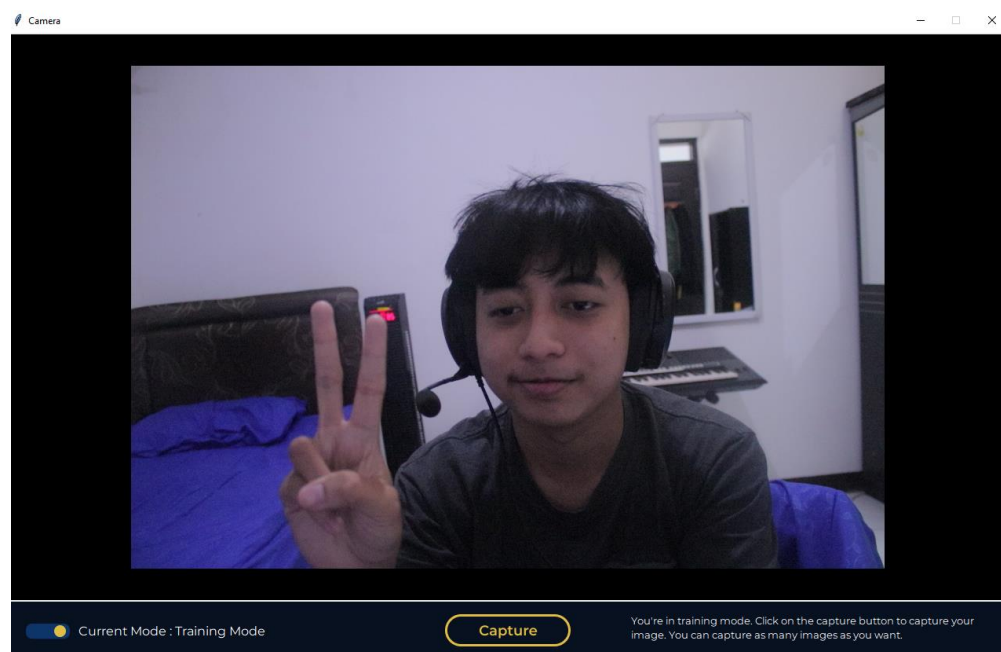
Gambar 3.1 Hasil Tes Face Recognition Dengan Sampel Citra Wajah Keanu Reeves

2. Contoh Image Reconstruction dengan Eigenface



Gambar 3.2 Hasil Rekonstruksi Gambar dengan Eigenface

3. Fitur Input melalui Kamera





Gambar 3.3 Hasil Input Melalui Kamera

BAB 4

EKSPERIMEN

1. Eksperimen berdasarkan variasi ukuran gambar

Pada eksperimen 4.1, akan diuji pengaruh ukuran training image terhadap akurasi *face recognition*. Pada eksperimen ini, akan diuji image berukuran 256x256 dan 512x512 dengan jumlah dataset 50. Test image tidak tepat berukuran 256x256 ataupun 512x512, akan tetapi mendekati, sehingga pada algoritma nantinya akan di-crop dan di-resize sesuai ukuran yang ingin di tes.

Ukuran Gambar	Hasil Eksperimen
1. 256 x 256 (ukuran dataset : 50) (varian orang : 5)	<div>1. Waktu eksekusi: 4.32 seconds</div> <div>Input Test Image: Emma Stone</div> <div></div> <div>Output Test Image: Emma Stone</div> <div></div> <div>2. Waktu eksekusi: 3.87 seconds</div> <div>Input Test Image: Zendaya</div>



Output Test Image:
Zendaya



3. Waktu eksekusi: 3.82 seconds

Input Test Image:
Mark Zuckeberg



Output Test Image:
Mark Zuckeberg



4. Waktu eksekusi: 4.02 seconds

Input Test Image:
Mark Zuckeberg



Output Test Image:
Zendaya



5. Waktu eksekusi: 3.90 seconds

Input Test Image:
Tom Holland



Output Test Image:
Zac Efron



6. Waktu eksekusi: 3.82 seconds

Input Test Image:
Random Dude



Output Test Image:
Mark Zuckerberg



7. Waktu eksekusi: 3.87 seconds
Input Test Image:
Random Dude



Output Test Image:
Mark Zuckerberg



8. Waktu eksekusi: 3.82 seconds
Input Test Image:
Zac Efron



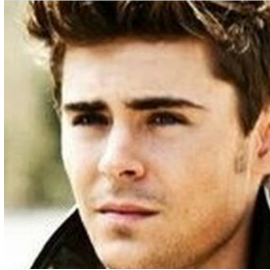
Output Test Image:
Tom Holland



9. Waktu eksekusi: 3.80 seconds

Input Test Image:

Zac Efron



Output Test Image:

Emma Stone



10. Waktu eksekusi: 4.12 seconds

Input Test Image:

Zendaya







Output Test Image:

Zac Efron



Analisis:

Berdasarkan hasil face recognition program di atas, dapat terlihat bahwa akurasi dari pengenalan wajah masih rendah. Menurut penulis, hal tersebut terjadi karena pengaruh posisi wajah dan kontras pencahayaan pada gambar.

	<p>Wajah-wajah yang memiliki kontras pencahayaan yang sama ataupun pose yang mirip akan cenderung dianggap sebagai <i>closest result</i>, karena program hanya menilai gambar berdasarkan nilai kecerlangan ketika image dibaca dalam warna <i>grayscale</i> (sehingga kontras cahaya lah yang lebih bermain di sini)</p>
<p>2. 512 x 512 (ukuran dataset : 50) (varian orang : 5)</p> <p>dataset sama dengan yang sebelumnya, tetapi sudah di-resize</p>	<p>1. Waktu eksekusi: 20.87 seconds Input Test Image: Emma Stone</p>  <p>Output Test Image: Emma Stone</p>  <p>2. Waktu eksekusi: 20.62 seconds Input Test Image: Zendaya</p>  <p>Output Test Image: Zendaya</p>  <p>3. Waktu eksekusi: 19.90 seconds Input Test Image: Mark Zuckerberg</p>



Output Test Image:
Mark Zuckerberg



4. Waktu eksekusi: 20.42 seconds

Input Test Image:
Mark Zuckerberg



Output Test Image:
Zendaya



5. Waktu eksekusi: 20.10 seconds

Input Test Image:
Tom Holland



Output Test Image:
Zac Efron



6. Waktu eksekusi: 20.10 seconds

Input Test Image:
Random Dude



Output Test Image:
Mark Zuckerberg



7. Waktu eksekusi: 20.10 seconds

Input Test Image:
Random dude



Output Test Image:
Mark Zuckerberg



8. Waktu eksekusi: 20.33 seconds

Input Test Image:

Zac Efron



Output Test Image:

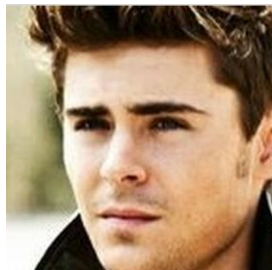
Tom Holland



9. Waktu eksekusi: 19.89 seconds

Input Test Image:



Zac Efron



Output Test Image:

Emma Stone



	<p>10. Waktu eksekusi: 20.05 seconds</p> <p>Input Test Image: Zendaya</p>  <p>Output Test Image: Zac Efron</p> 
	<p>Analisis:</p> <p>Dengan data yang sama dengan percobaan sebelumnya (hanya diubah sizenya saja), didapatkan hasil rekognisi wajah yang sama persis. Variabel yang terpengaruh oleh ukuran image hanyalah waktu eksekusi saja, sedangkan hasil tidak berpengaruh sama sekali.</p> <p>Untuk masalah akurasi, maka analisis tidak jauh berbeda dengan eksperimen 4.1.1, ketika meninjau gambar 256x256</p>

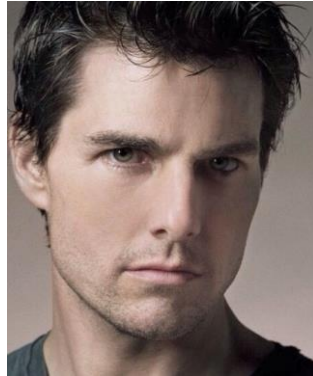
2. Eksperimen berdasarkan ukuran dataset

Pada eksperimen 4.2, akan diuji pengaruh jumlah training image terhadap akurasi *face recognition*. Pada eksperimen ini, akan diuji dataset berukuran 50, 100, 200, dan 300 gambar yang akan dibagi rata terhadap 5 orang.

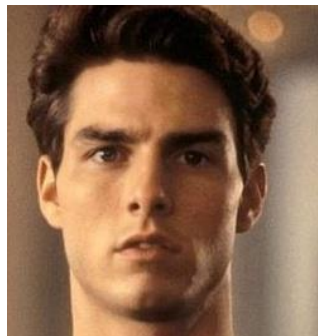
Ukuran Dataset	Hasil Eksperimen
50 gambar (10 gambar per orang)	1. Waktu eksekusi: 6,8 detik Input Test Image: Tom Holland



Output Test Image:
Tom Cruise



2. Waktu eksekusi: 6,55 detik
Input Test Image:
Tom Cruise



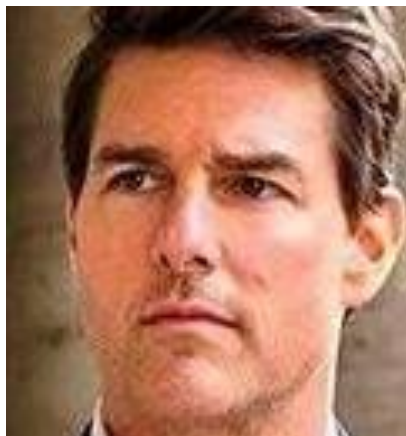
Output Test Image:
Tom Holland



3. Waktu eksekusi: 6,65
Input Test Image:
Natalie Dormer



Output Test Image:
Tom Cruise



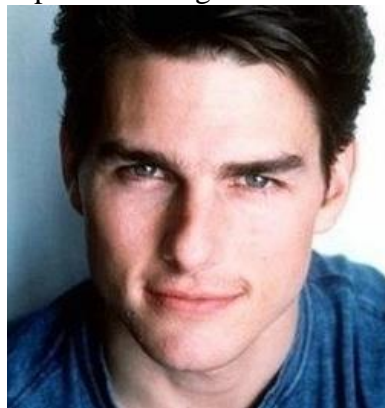
4. Waktu eksekusi: 6,65
Input Test Image:
Anne Hathaway



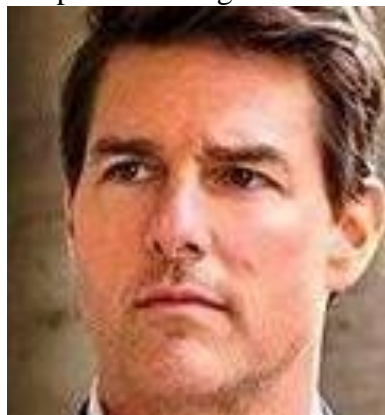
Output Test Image:
Tom Holland



5. Waktu eksekusi: 19 detik
Input Test Image: Tom Cruise



Output Test Image: Tom Cruise



6. Waktu eksekusi: 6,74
Input Test Image: Tom Holland



Output Test Image: Tom Holland



7. Waktu eksekusi: 6.61

Input Test Image: Natalie Dormer



Output Test Image: Tom Cruise



8. Waktu eksekusi: 12.87

Input Test Image: Keanu Reeves

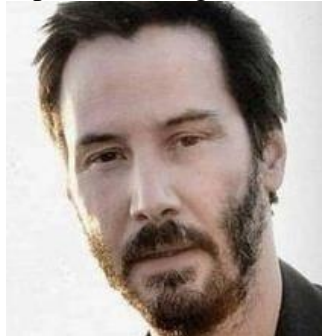


Output Test Image: Anne Hathaway



9. Waktu eksekusi 6,55 detik

Input Test Image: Keanu Reeves



Output Test Image: Tom Holland



10. Waktu eksekusi: 6.56

Input Test Image: Anne Hathaway

	<div data-bbox="603 208 956 577" data-label="Image"> </div> <div data-bbox="596 573 1053 616" data-label="Caption"> <p>Output Test Image: Keanu Reeves</p> </div> <div data-bbox="603 611 938 972" data-label="Image"> </div> <div data-bbox="596 1068 1374 1256" data-label="Text"> <p>Analisis: Dari tes dengan 50 dataset training, didapatkan 2 dari 10 gambar mengembalikan wajah orang yang sama. Akurasi tersebut sangatlah rendah Range waktu execution time berkisar di antara 6-7 detik dengan 2 <i>outlier</i>, yaitu 19 detik dan 12,87 detik.</p> </div>
<p>100 gambar (20 gambar per orang)</p>	<p>1. Waktu eksekusi: 29,12 detik Input Test Image: Tom Cruise</p> <div data-bbox="603 1357 904 1675" data-label="Image"> </div> <p>Output Test Image: Anne Hathaway</p>

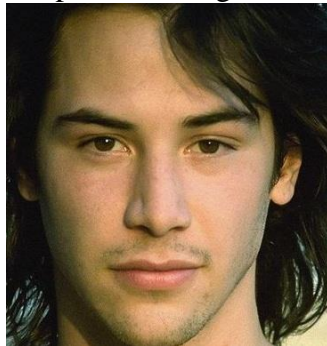


2. Waktu eksekusi: 55,47

Input Test Image: Anne Hathaway



Output Test Image: Keanu Reeves



3. Waktu eksekusi: 88,95

Input Test Image: Anne Hathaway



Output Test Image: Anne Hathaway



4. Waktu eksekusi: 92,19 detik
Input Test Image: Keanu Reeves



Output Test Image: Keanu Reeves



5. Waktu eksekusi: 89,98 detik
Input Test Image: Keanu Reeves



Output Test Image: Tom Holland



6. Waktu eksekusi: 28,92

Input Test Image: Tom Cruise



Output Test Image: Tom Holland



7. Waktu eksekusi: 88,82 detik

Input Test Image: Natalie Dormer



Output Test Image: Natalie Dormer



8. Waktu eksekusi: 80,61 detik
Input Test Image: Natalie Dormer



Output Test Image: Anne Hathaway



9. Waktu eksekusi: 77,98 detik
Input Test Image: Tom Holland



Output Test Image: Natalie Dormer



10. Waktu eksekusi: 87,22 detik
Input Test Image: Tom Holland



Output Test Image: Tom Holland



Analisis: Dari data tes 100 data training, didapatkan 4 dari 10 dataset mengembalikan gambar orang yang sama. Range runtime program berkisar di antara 70-90 detik dengan 2 outlier, yaitu 28,92 detik dan 55,47 detik.

200 gambar
(40 gambar per

1. Waktu eksekusi: 380,96
Input Test Image: Anne Hathaway

orang)



Output Test Image: Anne Hathaway



2. Waktu eksekusi:
Input Test Image:

Output Test Image:

3. Waktu eksekusi:
Input Test Image:

Output Test Image:

4. Waktu eksekusi:
Input Test Image:

Output Test Image:

5. Waktu eksekusi:
Input Test Image:

Output Test Image:

6. Waktu eksekusi:

	Input Test Image: Output Test Image: 7. Waktu eksekusi: Input Test Image: Output Test Image: 8. Waktu eksekusi: Input Test Image: Output Test Image: 9. Waktu eksekusi: Input Test Image: Output Test Image: 10. Waktu eksekusi: Input Test Image: Output Test Image:
	Analisis

BAB 5

KESIMPULAN, SARAN, DAN REFLEKSI

5.1 KESIMPULAN

Salah satu penerapan nilai eigen dan vektor eigen pada matriks adalah teknologi pengenalan wajah (*face recognition*) dengan metode eigenface. Algoritma eigenface diterjemahkan ke dalam Bahasa python dan menghasilkan suatu aplikasi pengenalan wajah. Aplikasi ini membutuhkan kumpulan-kumpulan gambar latih (*training image*). Gambar-gambar menjadi kumpulan-kumpulan matriks yang selanjutnya akan diproses. Matriks-matriks tersebut kemudian akan berukuran lebih kecil, tetapi tetap bisa mewakili gambar-gambar dasar tadi.

5.2 SARAN

1. Angle foto harus disetarakan saat melakukan tes

5.3 REFLEKSI

Pelajaran yang bisa diambil selama pengerjaan tugas besar 2 Algoritma Linier dan Geometri, yaitu :

1. Wawasan baru mengenai teknologi pengenalan wajah dan cara membuatnya,
2. Memperdalam penguasaan bahasa python,
3. Melatih manajemen waktu,
4. Melatih kerja sama dan komunikasi antar-anggota kelompok,

dan banyak lagi. Namun, empat poin tersebutlah yang menjadi hikmah utama dalam pengerjaan tugas besar kali ini.

REFERENSI

1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian2-2022.pdf>
2. <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf>
3. <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>
4. https://drive.google.com/file/d/1Qxxg_M69foCOBkh6Qr9aRTrwC86zK_rH/view
5. <https://www.youtube.com/watch?v=FAAnNBw7d0vg>

6. <https://www.youtube.com/watch?v=tYqOrvUOMFc>
7. <https://en.wikipedia.org/wiki/Eigenface>
8. <https://pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/>
9. <https://www.neliti.com/publications/135138/pengenalan-wajah-menggunakan-algoritma-eigenface-dan-euclidean-distance>

LAMPIRAN

Gambar