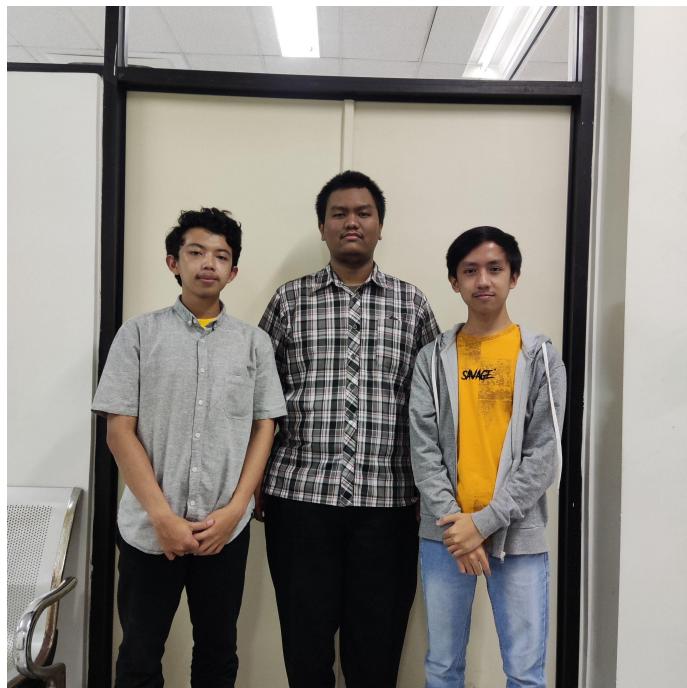


Laporan Tugas Besar 2

IF 2123 Aljabar Linier dan Geometri

Aplikasi Nilai Eigen dan EigenFace pada Pengenalan Wajah (Face Recognition)



Kelompok 46 (Face-X)

Ghazi Akmal Fauzan 13521058

Akmal Mahardika Nurwahyu P. 13521070

Ahmad Ghulam Ilham 13521118

Semester I Tahun 2022/2023

Sekolah Teknik Elektro dan Informatika

Jurusan Teknik Informatika

Institut Teknologi Bandung

BAB 1 Deskripsi Masalah

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.

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.

BAB 2

Teori Singkat

Perkalian Matriks

Perkalian matriks adalah suatu operasi dua matriks yang menghasilkan sebuah matriks. Agar dua matriks dapat dikalikan, banyaknya kolom pada matriks pertama harus sama dengan banyaknya baris pada matriks kedua. Matriks hasil perkalian keduanya, akan memiliki baris sebanyak baris matriks pertama, dan kolom sebanyak kolom matriks kedua.

Jika A adalah matriks berukuran $m \times n$ dan B adalah matriks berukuran $n \times p$, dengan elemen-elemen sebagai berikut,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

Hasil perkalian kedua matriks tersebut, $\mathbf{C} = \mathbf{AB}$, adalah sebuah matriks berukuran $m \times p$.

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

Dengan setiap entri pada matriks C didefinisikan sebagai

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^n a_{ik}b_{kj},$$

Matriks $\mathbf{C} = \mathbf{AB}$ dapat juga dituliskan sebagai

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

Nilai & Vektor Eigen

Kata “eigen” berasal dari Bahasa Jerman yang artinya “asli” atau “karakteristik”. Dengan kata lain, nilai eigen menyatakan nilai karakteristik dari sebuah matriks yang berukuran $n \times n$. Jika A adalah matriks $n \times n$ maka vektor tidak-nol x di R^n disebut vektor eigen dari A jika Ax sama dengan perkalian suatu skalar dengan x , yaitu $Ax = \lambda x$. Skalar λ disebut nilai eigen dari A , dan x dinamakan vektor eigen yang berkoresponden dengan λ .

Vektor eigen x menyatakan matriks kolom yang apabila dikalikan dengan sebuah matriks $n \times n$ menghasilkan vektor lain yang merupakan kelipatan vektor itu sendiri. Dengan kata lain, operasi $Ax = x$ menyebabkan vektor x menyusut atau memanjang dengan faktor dengan arah yang sama jika positif dan arah berkebalikan jika negatif.

Jika diberikan sebuah matriks A berukuran $n \times n$. Vektor eigen dan nilai eigen dari matriks A dihitung sebagai berikut:

$$\begin{aligned} Ax &= \lambda x \\ IAx &= \lambda Ix \quad (\text{kalikan kedua ruas dengan } I = \text{matriks identitas}) \\ Ax &= \lambda x \\ (\lambda I - A)x &= 0 \end{aligned}$$

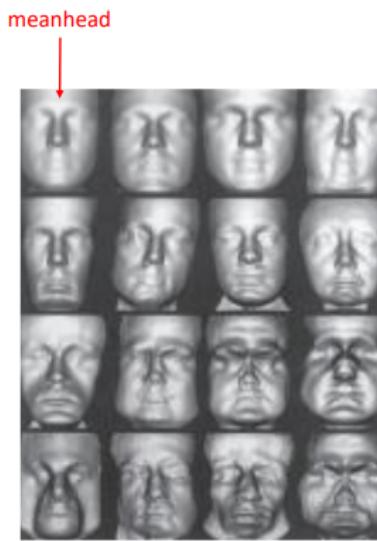
$x = 0$ adalah solusi trivial dari $(\lambda I - A)x = 0$. Agar $(\lambda I - A)x = 0$ memiliki solusi tidak-nol, maka haruslah $\det(\lambda I - A) = 0$. Persamaan $\det(\lambda I - A) = 0$ disebut persamaan karakteristik dari matriks A , dan akar-akar persamaan tersebut, yaitu λ , dinamakan akar-akar karakteristik atau nilai-nilai eigen.

Eigenface

Dikembangkan oleh Sirovich and Kirby dan digunakan oleh Matthew Turk and Alex Pentland untuk klasifikasi wajah, Eigenface adalah metode pengenalan wajah (face recognition) di berbasis vektor eigen dan nilai eigen yang digunakan untuk persoalan-persoalan di dalam computer vision. Vektor eigen diturunkan 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. Gagasananya adalah bahwa setiap wajah manusia dalam kelompok ras adalah kombinasi dari beberapa lusin bentuk primer.

Misalnya, dengan menganalisis pemindaian tiga dimensi dari banyak wajah, para peneliti di Universitas Rockefeller telah menghasilkan bentuk kepala rata-rata pada ras Kaukasia, dijuluki meanhead (gambar pojok kiri atas), dan satu himpunan berisi variasi wajah dari bentuk itu, yang disebut eigenheads (15 di antaranya ditunjukkan pada gambar).



Dinamakan eigenface karena merupakan vektor eigen dari matriks kovarian yang menyimpan informasi citra wajah. Bentuk wajah direpresentasikan secara matematis sebagai kombinasi linier dari eigenheads.



Algoritma 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 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.

I. Menyiapkan data training image

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

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

II. Menghitung nilai tengah data training image

Langkah kedua adalah ambil nilai tengah atau mean (Ψ).

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

III. Menghitung selisih antara nilai training image dan nilai tengah

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

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

IV. Menghitung matriks kovarian

Langkah keempat adalah menghitung nilai matriks kovarian (C).

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

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

V. Menghitung nilai eigen dan vektor eigen

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

$$C \times vi = \lambda i \times vi \quad (5)$$

VI. Menghitung eigenface

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

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

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

Tahapan Pengenalan wajah :

I. Menerapkan perhitungan eigenface pada test face

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

$$\mu_{new} = v \times \Gamma_{new} - \Psi \quad (7)$$

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

II. Menghitung euclidean distance antara test face dan masing-masing training image dalam database

Gunakan metode euclidean distance untuk mencari jarak (distance) terpendek antara nilai eigen dari training image dalam database dengan nilai eigen dari image testface.

$$\varepsilon_k = \Omega - \Omega_k \quad (8)$$

Tahapan Akhir

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.

BAB 3

Implementasi Program dan Kakas

I. Initialization

Nama	Parameter	Deskripsi
readImage	Input: Path Output: Matriks	Fungsi readImage menerima input path dari file training image. Akan dibuat suatu matriks berisi vektor-vektor training image dengan menggunakan library OpenCV. Fungsi mengembalikan matriks training image.
updateDatabase	Output: Database	Fungsi updateDatabase akan memanggil fungsi readImage untuk mendapatkan matriks penyimpan vektor-vektor training image. Akan dibuat database untuk menyimpan matriks training image. Fungsi mengembalikan database berisi matriks training image.
totalImage	Output: Integer	Fungsi totalImage akan membuka path berisi file training image. Akan dihitung banyak file training image yang disimpan dalam folder. Fungsi mengembalikan count (banyak file training image).

Spesifikasi dan Deskripsi Fungsi dan Prosedur initialization.py

II. Mean

Nama	Parameter	Deskripsi
mean	Output: Matriks	<p>Fungsi mean membuka database untuk mencari matriks penyimpan vektor-vektor training image. Akan dihitung matriks mean dengan menjumlahkan tiap vektor training image dan membaginya dengan banyak vektor training image.</p> <p>Fungsi mengembalikan matriks berisi mean dari matriks training image.</p>

Spesifikasi dan Deskripsi Fungsi dan Prosedur mean.py

III. Selisih

Nama	Parameter	Deskripsi
selisih	Output: Database	<p>I.S. : Matriks training image terdefinisi dan tersimpan di database.</p> <p>Prosedur selisih akan membuka database untuk mencari matriks penyimpan vektor-vektor training image. Akan dihitung matriks selisih dengan mengurangkan matriks training image dengan matriks mean. Matriks selisih kemudian disimpan ke database.</p> <p>F.S. : Prosedur selisih mengupdate database matriks training image dengan matriks selisih.</p>

Spesifikasi dan Deskripsi Fungsi dan Prosedur selisih.py

IV. Kovarian

Nama	Parameter	Deskripsi
getCovariance	Input: Path Output: Matriks, Matriks	Fungsi covariance menerima input database yang berisi matriks selisih A. Fungsi akan menyusun matriks transpose A (AT). Akan dihitung matriks kovarian L sebagai hasil perkalian matriks AT dan matriks A. $L = AT * A$ Fungsi mengembalikan matriks kovarian L dan matriks selisih A.

Spesifikasi dan Deskripsi Fungsi dan Prosedur covariance.py

V. Eigen

Nama	Parameter	Deskripsi
getQR	Input: Matriks Output: Matriks, Matriks	Fungsi getQR menerima input matriks M. Fungsi melakukan proses pencarian matriks QR. $M = Q * R$ Fungsi mengembalikan matriks Q dan matriks R
getEigen	Input : Matriks, Matriks Output : Matriks	Fungsi getEigen menerima input matriks kovarian C dan matriks selisih A. Fungsi menggunakan metode dekomposisi QR untuk mencari vektor-vektor eigen dan nilai eigen dari matriks kovarian C.

		Fungsi mengembalikan matriks vektor-vektor eigen.
pickEigen	Input : Matriks Output : Matriks	<p>Fungsi menerima input matriks berisi vektor-vektor eigen dan memilih K vektor eigen terbaik yang sudah terurut berdasarkan nilai eigen.</p> <p>Fungsi mengembalikan matriks K vektor eigen terbaik.</p>
CalculateEigenFace	Input : Matriks	<p>I.S. : Matriks vektor eigen terdefinisi, matriks selisih terdefinisi</p> <p>Prosedur menerima input matriks berisi vektor-vektor eigen. Akan dilakukan kalkulasi untuk mencari matriks Eigenface dari matriks vektor eigen dan matriks selisih. Matriks eigenface akan disimpan di database</p> <p>F.S. : Prosedur mengupdate database penyimpanan Eigenface.</p>

Spesifikasi dan Deskripsi Fungsi dan Prosedur eigen.py

VI. Identification

Nama	Parameter	Deskripsi
getNewEigenFace	Input : Image, Path, Matriks	Fungsi getNewEigenFace menerima 3 input, yaitu image yang akan dites, path database Eigenface, dan matriks vektor eigen. Fungsi akan mencari selisih vektor testface dengan mean matriks training image dan mengalikan vektor selisih testface dengan matriks vektor eigen. Selanjutnya fungsi membuka database Eigenface dan menelusuri euclidean distance

		<p>terkecil antara vektor Eigenface testface dengan vektor-vektor di database Eigenface.</p> <p>Fungsi mengembalikan file yang memiliki euclidean distance terpendek dan nilai euclidean distance tersebut.</p>
--	--	---

Spesifikasi dan Deskripsi Fungsi dan Prosedur identification.py

VII. Video

Nama	Parameter	Deskripsi
start (Class VideoCapture)	Input : Video Capture	<p>I.S. : Objek foto dan ukuran foto sembarang.</p> <p>Prosedur start akan menginisialisasi pengambilan muka yang akan dites. Foto diambil menggunakan kamera dan library OpenCV.</p> <p>F.S. : Objek foto dan ukuran foto terdefinisi.</p>
Stop (Class VideoCapture)	Input : Video Capture	<p>I.S. : Video capture dijalankan, objek foto terdefinisi.</p> <p>Prosedur stop akan mengecek status video capture. Jika pengguna sudah selesai menggunakan video capture, prosedur akan mengubah status video capture.</p> <p>F.S. : Video capture dihentikan, objek foto tidak terdefinisi.</p>

get_image	Input : Video Capture	<p>Fungsi get_image membaca status video capture. Jika objek foto terdeteksi dan valid, prosedur membuat image yang akan dijadikan sebagai testface untuk pengecekan Eigenface.</p> <p>Fungsi mengembalikan image testface jika berhasil dan mengembalikan boolean False jika tidak berhasil.</p>
-----------	-----------------------	---

Spesifikasi dan Deskripsi Fungsi dan Prosedur video.py

VIII. Main

Nama	Parameter	Deskripsi
check_entry		<p>I.S. : Dataset dan foto sembarang Prosedur check_entry akan memanggil fungsi get untuk mengecek dataset dan foto. Jika tidak ada dataset dan foto yang diinput, program utama tidak akan mengeluarkan apa pun.</p> <p>F.S. : Jika dataset dan foto terdeteksi dan valid, dataset dan foto terdefinisi.</p>
resize	Input : Image	<p>Fungsi resize menerima input image yang akan dites. Fungsi menyesuaikan ukuran foto testface agar perhitungan dapat dilakukan dengan mudah.</p> <p>Fungsi resize mengembalikan foto yang ukurannya sudah disesuaikan.</p>

select_dataset		<p>I.S. : Database sembarang.</p> <p>Prosedur select_dataset memanggil fungsi filedialog.askdirectory untuk menentukan apakah path sudah terdefinisi. Jika path terdefinisi, proses akan dilanjutkan ke program utama.</p> <p>F.S. : Jika path ditemukan, dataset terdefinisi.</p>
select_image		<p>I.S. : Image sembarang.</p> <p>Prosedur select_image memanggil fungsi filedialog.askopenfilename untuk menentukan apakah file image sudah terdefinisi. Jika file image terdefinisi, proses akan dilanjutkan ke program utama.</p> <p>F.S. : Jika file image ditemukan, image terdefinisi.</p>
update_status	Input : Status (String)	<p>I.S. : Status awal terdefinisi</p> <p>Prosedur update_status akan menyesuaikan tampilan status setiap ada perubahan terhadap status</p> <p>F.S. : Status akhir terdefinisi, bisa berbeda dengan status awal</p>
update_dataset		<p>I.S. : Path dataset terdefinisi</p> <p>Prosedur akan melakukan pengecekan apakah dataset sudah terupdate atau belum. Apabila belum, prosedur update dataset akan melakukan update terhadap dataset yang lama.</p>

		F.S. : Dataset diperbarui.
stopwatch		<p>I.S. : Waktu proses sembarang</p> <p>Prosedur stopwatch menampilkan waktu pemrosesan testface dengan menggunakan library time.</p> <p>F.S. : Waktu proses ditampilkan di GUI.</p>
start_process		<p>I.S. : Proses program utama belum dimulai.</p> <p>Prosedur start_process akan mengecek path dataset dan input image. Jika path dan image input terdefinisi, proses program utama akan dimulai. Jika tidak, akan menampilkan pesan error.</p> <p>F.S. : Jika dataset dan image input terdefinisi, memulai proses program utama. Jika tidak, menampilkan pesan error.</p>
stop_video		<p>I.S. : Video capture masih digunakan.</p> <p>Prosedur stop_video akan dipanggil jika pengguna tidak ingin menggunakan fitur video capture lagi.</p> <p>F.S. : Video capture berhenti digunakan.</p>
start_video		<p>I.S. : Video capture belum digunakan, foto belum terdefinisi.</p> <p>Prosedur start_video terlebih dahulu mengecek apakah database sudah tersedia. Jika sudah, prosedur akan mulai merekam foto hasil video capture dan menyimpannya.</p>

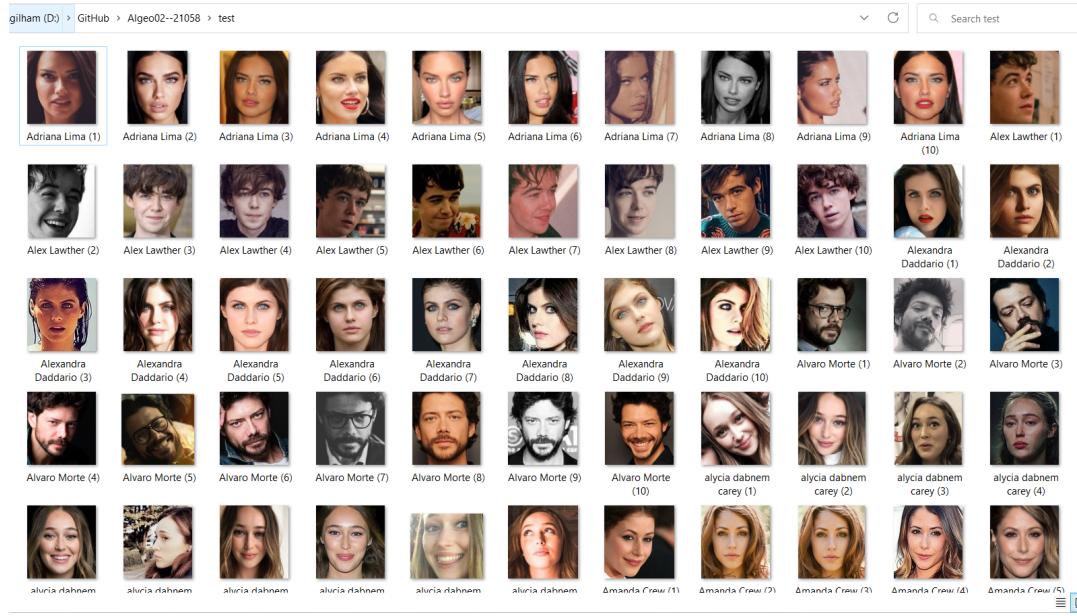
		<p>Hasil rekaman akan digunakan untuk pemrosesan program utama.</p> <p>F.S. : Jika database terdefinisi, video capture digunakan dan foto terdefinisi.</p>
start		<p>I.S. : Program belum dimulai.</p> <p>Prosedur start akan memulai program dan memanggil fungsi dan prosedur yang sudah diimplementasikan.</p> <p>F.S. : Program dimulai, pemrosesan siap dilakukan.</p>
close		<p>I.S. : Program masih berjalan.</p> <p>Prosedur close akan digunakan jika pengguna ingin berhenti menjalankan program.</p> <p>F.S. : Program berhenti dijalankan.</p>

Spesifikasi dan Deskripsi Fungsi dan Prosedur main.py

BAB 4

Eksperimen

I. Initialization



Path Folder Training Image

```
Matriks 1x2048 Gambar  
[ 0.46099806  0.51584506  0.46215373 ... -0.00936169  0.09766258  
 0.09397753 ]
```

Hasil Fungsi / Prosedur readImage

```

Isi database
{'test\\adriana lima (1).jpg': array([ 0.46899886,  0.51584506,  0.46215373, -0.0936169,
  0.04872968], dtype=float32), 'test\\adriana lima (10).jpg': array([ 0.06211631,  0.03774853,  0.07521775, ...,  0.02512084,  0.02951337,
  0.0919039 ], dtype=float32), 'test\\adriana lima (2).jpg': array([-0.01862869,  0.01870345,  0.06339239, ...,  0.02177417,  0.0664984 , 0.05922471,  0.04999999], dtype=float32), 'test\\adriana lima (3).jpg': array([-0.00167622, -0.00035351,  0.00028184, ..., -0.02155226, 0.05926879,  0.04668445], dtype=float32), 'test\\adriana lima (4).jpg': array([-0.00167622,  0.02549949, -0.0020552 ,  0.02545985, ...,  0.02597876, 0.06631527,  0.07192404], dtype=float32), 'test\\adriana lima (5).jpg': array([-0.0056549 ,  0.00218093,  0.00798487, ...,  0.04557681, 0.03992747,  0.14786364], dtype=float32), 'test\\adriana lima (6).jpg': array([-0.01223871,  0.00316615,  0.01688473, ...,  0.03655294, 0.0806573 ,  0.05310293], dtype=float32), 'test\\adriana lima (7).jpg': array([-0.1424816 , -0.13112979,  0.14918138, ...,  0.03013118, 0.09962198], dtype=float32), 'test\\adriana lima (8).jpg': array([-0.01209339,  0.02484305,  0.05753046, ...,  0.048580839,  0.06209866, 0.08536414,  0.06678986], dtype=float32), 'test\\adriana lima (9).jpg': array([-0.15020629,  0.078494296,  0.16297176, ..., -0.08447427, 0.07826751], dtype=float32), 'test\\alex lawther (10).jpg': array([-0.00966899,  0.00379438,  0.06499358, ...,  0.02295856, 0.06913996,  0.02323977], dtype=float32), 'test\\alex lawther (2).jpg': array([-0.00196826, -0.00417893,  0.00447011, ...,  0.00783704, 0.03418804,  0.03499828], dtype=float32), 'test\\alex lawther (3).jpg': array([-0.00297069,  0.00131325,  0.00361664, ...,  0.16385564, 0.05794543,  0.12018379], dtype=float32), 'test\\alex lawther (4).jpg': array([-0.0716589 ,  0.03015952,  0.07676894, ..., -0.00130795, 0.00388626,  0.00169673], dtype=float32), 'test\\alex lawther (5).jpg': array([-0.02001392,  0.02374732,  0.095728 , 0.01012207,  0.003086461, 0.18102693], dtype=float32), 'test\\alex lawther (6).jpg': array([-0.01052884,  0.00639139,  0.01301601, ...,  0.0731101 , 0.06726734,  0.08937679], dtype=float32), 'test\\alex lawther (7).jpg': array([-0.001380892,  0.01010343,  0.06413474, ..., -0.00880233, 0.089424887,  0.004674894], dtype=float32), 'test\\alex lawther (8).jpg': array([-0.16128268, -0.00932427,  0.21585518, ..., -0.00147585, 0.05372193,  0.064992978], dtype=float32), 'test\\alex lawther (9).jpg': array([-0.04386555,  0.004072575,  0.07340831, ..., -0.02137434, 0.15114683,  0.0697109 ], dtype=float32), 'test\\alexandra daddario (1).jpg': array([-0.06108854, -0.07461435,  0.07008858, ..., -0.01860061, 0.03743543,  0.020866021], dtype=float32), 'test\\alexandra daddario (10).jpg': array([-0.23647742, -0.21553287,  0.24058746, ...,  0.00566891, 0.12941994,  0.03361747], dtype=float32), 'test\\alexandra daddario (2).jpg': array([-0.00122479,  0.00063777,  0.00130192, ...,  0.0267568 , 0.06198001,  0.05611406], dtype=float32), 'test\\alexandra daddario (3).jpg': array([-3.72108668e-03, -8.3575994e-02,  1.2893492e-02, ..., -0.00010001, 0.1788666e-05,  1.9976852e-03,  1.6566054e-03], dtype=float32), 'test\\alexandra daddario (4).jpg': array([-0.02053547, -0.06353186,  0.05387704, ...,  0.03156399, 0.08334996,  0.03984402], dtype=float32), 'test\\alexandra daddario (5).jpg': array([-0.18422654,  0.02413235,  0.25948685, ...,  0.08721535, 0.11064541, 0.09431594], dtype=float32), 'test\\alexandra daddario (6).jpg': array([-0.00827539,  0.05182401, ...,  0.00941524], 0.07855552,  0.050822789], dtype=float32), 'test\\alexandra daddario (7).jpg': array([-0.07737223,  0.04102542,  0.07793991, ..., -0.00466149, 0.0768636 ,  0.017841176], dtype=float32), 'test\\alexandra daddario (8).jpg': array([-0.00368265,  0.00291265,  0.00795304, ..., -0.00401203, 0.05629943,  0.06353886], dtype=float32), 'test\\alexandra daddario (9).jpg': array([-0.00559983, -0.00677154,  0.01687901, ...,  0.00709391, 0.01985821,  0.04983731], dtype=float32), 'test\\alexandra daddario (10).jpg': array([-0.0008301, -0.001512 ,  0.09327494, ...,  0.0510578 , 0.12244178,  0.08666005], dtype=float32), 'test\\alvano morte (1).jpg': array([-0.00011588, -0.00511521,  0.08535781, ..., -0.01693771, 0.12467246,  0.06667183], dtype=float32), 'test\\alvano morte (2).jpg': array([-0.00588097,  0.064116621,  0.108808476, ...,  0.05452606, 0.02908029,  0.05682157], dtype=float32), 'test\\alvano morte (3).jpg': array([-0.00932243,  0.02755822,  0.04983026, ...,  0.004664628, 0.01152175,  0.01145762], dtype=float32), 'test\\alvano morte (4).jpg': array([-0.36943454, -0.132101495,  0.535764046, ..., -0.026624115, 0.01624419,  0.03172561], dtype=float32), 'test\\alvano morte (5).jpg': array([-0.00528945,  0.006250452,  0.01224373, ..., -0.6425696 , 0.07917518,  0.64007333], dtype=float32), 'test\\alvano morte (6).jpg': array([-0.24821994,  0.06339278,  0.241401106, ..., -0.82968855, 0.12018748,  0.08583757], dtype=float32), 'test\\alvano morte (7).jpg': array([-0.001614 ,  0.00113235,  0.00196987, ...,  0.02242535, 0.08277186,  0.8396479 ], dtype=float32), 'test\\alvano morte (8).jpg': array([-0.01183182, -0.05115196,  0.62018713, ..., -0.062339463, 0.053383 ,  0.07753862], dtype=float32), 'test\\alvano morte (9).jpg': array([-0.0889678 ,  0.0386304 ,  0.13103211, ...,  0.01518805,  0.02232593, 0.08522198], dtype=float32), 'test\\alycia dabnem carey (1).jpg': array([-0.01607392, -0.03323755,  0.03562369, ..., -0.02374491, 0.13387923,  0.07266383], dtype=float32), 'test\\alycia dabnem carey (10).jpg': array([-0.00284835, -0.01848876,  0.02032259, ..., -0.05075723, 0.19355483,  0.12956582], dtype=float32), 'test\\alycia dabnem carey (2).jpg': array([-0.00550287, -0.03295713,  0.03063642, ...,  0.00959744, 0.01946823,  0.03210852], dtype=float32), 'test\\alycia dabnem carey (3).jpg': array([-0.01827278, -0.000462078,  0.00486417, ...,  0.00850593, 0.05285824,  0.12612093], dtype=float32), 'test\\alycia dabnem carey (4).jpg': array([-0.866333613,  0.04727812,  0.11382335, ...,  0.008848425,  0.07847971 , 0.06977379], dtype=float32), 'test\\alycia dabnem carey (5).jpg': array([-0.0037135 ,  0.12005103, -0.0479984 , 0.0765982 ,  0.14512841], dtype=float32), 'test\\alycia dabnem carey (6).jpg': array([-0.00846377, -0.02291409,  0.02557122, ...,  0.00853203, 0.03349595,  0.03274617], dtype=float32), 'test\\alycia dabnem carey (7).jpg': array([-0.11325141, -0.09117211,  0.11531968, ...,  0.00932485, 0.12119824,  0.02427813], dtype=float32), 'test\\alycia dabnem carey (8).jpg': array([-0.00668607,  0.01957912,  0.02627621, ..., -0.01171908, 0.04889049,  0.16587752], dtype=float32), 'test\\alycia dabnem carey (9).jpg': array([-0.00185916, -0.01842246,  0.10382729, ..., -0.01942685, 0.0995777 ,  0.280433787], dtype=float32)}

```

Hasil Fungsi / Prosedur updateDatabase



Hasil Fungsi / Prosedur totalImage

II. Mean

Matriks Mean
$-0.00310032 \quad 0.00090384 \quad 0.08820847 \quad \dots \quad -0.00098296 \quad 0.06532097$
$0.07893113]$

Hasil Fungsi / Prosedur mean

III. Selisih

```
Isi database selisih
{'test\\adriana lima (1).jpg': array([ 0.46409838,  0.51494122,  0.37394526, ..., -0.08837874,
   -0.03234161,  0.01504641]), 'test\\adriana lima (10).jpg': array([ 0.06521663,  0.03684469, -0.01299073, ...,  0.02610299,
   -0.0358076, -0.03020145]), 'test\\adriana lima (2).jpg': array([ 0.02172902,  0.01779961, -0.02481669, ...,  0.02275712,
   -0.08116943,  0.01297278]), 'test\\adriana lima (3).jpg': array([ 0.00477655, -0.00125736, -0.07992664, ..., -0.02056931,
   -0.08669626, -0.03394614]), 'test\\adriana lima (4).jpg': array([ 0.02859872, -0.00295984, -0.06217342, ..., -0.02498781,
   -0.08612017, -0.03224667]), 'test\\adriana lima (5).jpg': array([-0.08255458,  0.00129189, -0.08822404, ...,  0.04655976,
   -0.0856853, -0.00010673]), 'test\\adriana lima (6).jpg': array([-0.08913839,  0.00285627, -0.07132374, ...,  0.03753589,
   -0.0323405, -0.00010673]), 'test\\adriana lima (7).jpg': array([-0.13930074, -0.13930074,  0.06915781, ...,  0.03111513,
   -0.01533634,  0.02573282]), 'test\\adriana lima (8).jpg': array([ 0.0189071,  0.07363921, -0.07363921, ...,  0.03949134,
   -0.08322231,  0.01728872]), 'test\\adriana lima (9).jpg': array([ 0.15330861,  0.07463921,  0.07463921, ..., -0.0030131,
   -0.05995683, -0.07215826]), 'test\\alex laurther (1).jpg': array([ 0.05365007,  0.03889132,  0.05046084, ...,  0.05881806,
   -0.00235312, -0.00066361]), 'test\\alex laurther (10).jpg': array([-0.00650867,  0.00289054, -0.0232749, ...,  0.02394151,
   -0.05618101,  0.05570835]), 'test\\alex laurther (2).jpg': array([ 0.00113206, -0.08568277, -0.08373336, ...,  0.08881999,
   -0.03114051, -0.00403384]), 'test\\alex laurther (3).jpg': array([ 0.00013824,  0.00040941, -0.08459783, ...,  0.1648386,
   -0.00737553,  0.01125266]), 'test\\alex laurther (4).jpg': array([ 0.07475122,  0.02925568, -0.01143053, ..., -0.00032499,
   -0.06223471, -0.0772340]), 'test\\alex laurther (5).jpg': array([ 0.02311424,  0.02281348,  0.09751952, ...,  0.10118583,
   -0.06225635,  0.0222269]), 'test\\alex laurther (6).jpg': array([-0.00742852,  0.00548755, -0.07519246, ...,  0.07489376,
   -0.00191637,  0.01043966]), 'test\\alex laurther (7).jpg': array([ 0.06640892,  0.00911959, -0.02407373, ..., -0.00789937,
   -0.0239279, -0.03211821]), 'test\\alex laurther (8).jpg': array([-0.15872654, -0.05022811,  0.12684671, ..., -0.0004929,
   -0.01159904, -0.01400134]), 'test\\alex laurther (9).jpg': array([-0.04076524,  0.03982191, -0.01488016, ..., -0.02039138,
   -0.08582586, -0.08922822]), 'test\\alexander daddario (1).jpg': array([-0.05798586, -0.07551819, -0.01811989, ..., -0.01761765,
   -0.02788556, -0.021643671]), 'test\\alexander daddario (10).jpg': array([-0.2333771,  0.21643671,  0.15237898, ...,  0.00665187,
   -0.06409897, -0.04513166]), 'test\\alexander daddario (2).jpg': array([-0.00043251, -0.00154154, -0.086969655, ...,  0.02773975,
   -0.08334096, -0.02817066]), 'test\\alexander daddario (3).jpg': array([-0.000620796, -0.08447983, -0.07531498, ...,  0.00093117,
   -0.06332336, -0.07727452]), 'test\\alexander daddario (4).jpg': array([-0.01743515, -0.06443537, -0.03433144, ...,  0.03254695,
   -0.01892899, -0.03988711]), 'test\\alexander daddario (5).jpg': array([-0.18752686,  0.0152851,  0.17127838, ...,  0.0881983,  0.04532444,
   -0.01538601, -0.00517586]), 'test\\alexandra daddario (6).jpg': array([-0.00517586, -0.0152851,  0.08360626, ...,  0.000169819,
   -0.01538601, -0.00517586]), 'test\\alexandra daddario (7).jpg': array([-0.074627101,  0.000182157, -0.01925657, ..., -0.00367854,
   -0.01518263, -0.01608877]), 'test\\alexandra daddario (8).jpg': array([-0.08055233, -0.00208881, -0.00925544, ..., -0.03949134,
   -0.00903854, -0.01520232]), 'test\\alexandra daddario (9).jpg': array([-0.00700915,  0.00767539, -0.07132946, ...,  0.00007687,
   -0.08456275, -0.03989381]), 'test\\alvaro morte (1).jpg': array([-0.08393802, -0.05240288,  0.09506646, ...,  0.052801076,
   -0.05712081, -0.00766937]), 'test\\alvaro morte (10).jpg': array([-0.000425917, -0.00605129,  0.08285066, ..., -0.01595144,
   -0.05935149, -0.0128593]), 'test\\alvaro morte (2).jpg': array([-0.00198865,  0.06326257,  0.01987629, ...,  0.05559901,
   -0.03631897, -0.02189555]), 'test\\alvaro morte (3).jpg': array([-0.00622211,  0.02664638, -0.03837821, ...,  0.00561723,
   -0.05010347, -0.06747751]), 'test\\alvaro morte (4).jpg': array([-0.36633421, -0.13331429,  0.04755598, ..., -0.02504112,
   -0.04997678, -0.00728552]), 'test\\alvaro morte (5).jpg': array([-0.00838977,  0.00556868, -0.07596745, ..., -0.64158665,
   -0.01385421, -0.06554219]), 'test\\alvaro morte (6).jpg': array([-0.23711961,  0.06248894,  0.15323258, ..., -0.02878559,
   -0.05486651, -0.00696644]), 'test\\alvaro morte (7).jpg': array([-0.00148632,  0.00028251, -0.0862394, ...,  0.02340831,
   -0.01745089, -0.03928322]), 'test\\alvaro morte (8).jpg': array([-0.01943214, -0.05285581, -0.06810135, ..., -0.06132168,
   -0.01193797, -0.00140865]), 'test\\alvaro morte (9).jpg': array([-0.00206812,  0.03772656,  0.04282364, ...,  0.01616301,
   -0.04299504, -0.00629986]), 'test\\alycia dabnen carey (1).jpg': array([-0.01917424, -0.03414139, -0.05258478, ..., -0.02276195,
   -0.06855826, -0.00626733]), 'test\\alycia dabnen carey (10).jpg': array([-0.00594867, -0.0193848, -0.06388588, ..., -0.04977428,
   -0.12823387, -0.05063469]), 'test\\alycia dabnen carey (2).jpg': array([-0.00244254, -0.03386697, -0.05757285, ...,  0.01058639,
   -0.04585274, -0.04682234]), 'test\\alycia dabnen carey (3).jpg': array([-0.00872746, -0.00552463, -0.04734143, ...,  0.08864226,
   -0.01246273, -0.04719381]), 'test\\alycia dabnen carey (4).jpg': array([-0.06943645,  0.00437428,  0.02561488, ...,  0.0094672,
   -0.01315813, -0.00911573]), 'test\\alycia dabnen carey (5).jpg': array([-0.00061317,  0.01666218,  0.03184556, ..., -0.04781569,
   -0.06582725, -0.06191732]), 'test\\alycia dabnen carey (6).jpg': array([-0.01158403, -0.02381793, -0.06263726, ...,  0.00951099,
   -0.03182582, -0.00408496]), 'test\\alycia dabnen carey (7).jpg': array([-0.11825109, -0.09287595,  0.02711112, ...,  0.00130781,
   -0.05586927, -0.05045299]), 'test\\alycia dabnen carey (8).jpg': array([-0.00996639,  0.01866628, -0.06193826, ..., -0.01073613,
   -0.01641687, -0.00694659]), 'test\\alycia dabnen carey (9).jpg': array([-0.008495948,  0.0193263,  0.01561882, ..., -0.0184439,
   -0.03425673,  0.12540674])}]
```

Hasil Fungsi / Prosedur selisih

IV. Kovarian

```
Covmat
[[11.57541338 -1.17741994 -0.34373848 ... -0.5171926 -0.79814138
  0.04776331]
 [-1.17741994 11.19677275 0.61837394 ... -0.68619475 1.16373407
  0.23324845]
 [-0.34373848 0.61837394 7.83661646 ... -0.08620763 -0.12474035
  -1.25431143]
 ...
 [-0.5171926 -0.68619475 -0.08620763 ... 13.85007294 -0.41101356
  0.28555017]
 [-0.79814138 1.16373407 -0.12474035 ... -0.41101356 11.98054137
  -0.26046237]
 [ 0.04776331 0.23324845 -1.25431143 ... 0.28555017 -0.26046237
  15.75197015]]
```

Hasil Fungsi / Prosedur getCovariance

```

Diffmat
[[ 0.46409838  0.06521663  0.02172902 ... -0.11025109  0.00996039
  0.08495948]
 [ 0.51494122  0.03684469  0.01779961 ... -0.09207595  0.01866628
 -0.0193263 ]
 [ 0.37394526 -0.01299073 -0.02481609 ...  0.0271112  -0.06193826
  0.01561882]
...
[-0.00837874  0.02610299  0.02275712 ...  0.00130781 -0.01073613
 -0.0184439 ]
[ 0.03234161 -0.0358076   0.00116943 ...  0.05586927 -0.01641687
  0.03425673]
[ 0.01504641 -0.03020145  0.01297278 ... -0.05465299  0.08694639
  0.12540674]]
```

Hasil Fungsi / Prosedur getCovariance

V. Eigen

```

Eigenvector
[[ 0.45692882  0.12357337  0.09620381 ... -0.10665765 -0.00886358
  0.08850739]
 [ 0.51935891  0.10504939  0.02504724 ... -0.05163023  0.03122258
  0.13876906]
 [ 0.30637273  0.07978866 -0.09349281 ...  0.06282817 -0.05830076
  0.18187368]
...
[-0.03584195  0.0368503   0.02338264 ...  0.00847889  0.03208108
 -0.0241034 ]
[ 0.03159111 -0.03395454  0.00780746 ...  0.02936007 -0.01817623
 -0.02613492]
[ 0.05954738 -0.01693377  0.00492159 ... -0.13468714 -0.03520327
 -0.08740156]]
```

Hasil Fungsi / Prosedur getEigen

```

Isi database eigen
{'test\\adriana lima (1).jpg': array([ 1.24371829e+01, -1.64798730e-17, 4.34981912e-16, 5.68880880e-16,
-1.35525272e-18, 4.90384643e-16, -3.24393290e-16, 4.56232274e-16,
5.62484087e-16, 3.65159292e-16, -6.93889390e-16, -6.00431163e-16,
1.04842358e-16, -9.10512984e-16, -6.47919218e-16, 2.77555756e-17,
-3.24826971e-16, 2.88397778e-16, 1.09938100e-15, 2.25514852e-17,
-8.99887803e-17, 0.00000000e+00, -6.83481050e-16, 4.97865638e-16,
-5.59448321e-17, 1.99493200e-16, 3.59887760e-16, -2.61075883e-16,
-8.99887803e-17, 0.00000000e+00, -6.83481050e-16, 4.97865638e-16,
-5.59448321e-17, 1.99493200e-16, 3.59887760e-16, -2.61075883e-16,
-2.12563626e-16, 2.85362612e-16, 3.86843335e-16, -1.07552856e-16,
5.79831322e-16, -5.36829554e-16, 4.95263552e-16, -3.06612374e-16,
2.872759927e-16, 5.09141340e-16, 1.24488893e-15, 1.03763569e-15,
5.531789948e-16, 3.61256164e-16, 3.46944695e-18, 1.14091749e-15,
1.84704682e-15, 1.25264909e+00], 'test\\adriana lima (10).jpg': array([-2.52525345e+00, 1.18034116e+01, -4.93095148e-16, -3.02926087e-16,
-1.68502550e-16, -3.42909946e-16, 9.54825507e-17, -5.37760278e-17,
-4.70411167e-16, -1.06350031e-16, 3.85922774e-16, 1.03572770e-16,
2.18335221e-17, 1.33572770e-16, 1.56992478e-16, 6.08887930e-16,
-1.96023753e-16, 3.50475690e-16, 2.31266323e-16, 2.56770974e-16,
5.20615131e-16, 3.0375320e-16, 8.8217497e-16, 1.40323894e-16,
1.18848830e-15, -1.85818132e-15, 3.68347899e-16, 1.42897816e-16,
5.85035492e-16, 2.619417245e-16, 8.49147141e-16, 6.646322772e-16,
-3.80107161e-16, 9.37925620e-16, 3.75266652e-16, 6.66133815e-16,
6.37518877e-16, 7.87138777e-16, 9.17668719e-16, 1.18828558e-16,
1.34007389e-16, 3.18515502e-16, -2.40549608e-16, 1.67856188e-15,
4.53888888e-16, 8.31799907e-16, 2.33458012e-15, 2.69229883e-15,
5.53783236e-15, 2.16922188e+00], 'test\\adriana lima (2).jpg': array([-6.76291197e-01, 1.39373681e+00, 8.64283292e+00, -9.09428782e-16,
-1.39157349e-16, -2.56739704e-16, -1.54542178e-15, -1.50853581e-16,
1.55257751e-16, -6.23416249e-16, 3.30247982e-16, -3.99562563e-16,
-2.287413562e-16, -6.72285347e-17, 8.91211186e-17, 4.97226336e-16,
-2.82759927e-16, -1.30184261e-17, 6.17995238e-17, -4.91791105e-16,
-6.16477355e-16, -5.03869888e-16, -7.82368288e-16, 3.857458013e-17,
-4.55364912e-17, 3.58414142e-16, 1.684858018e-16, 3.32858067e-16,
-1.09894732e-15, 2.11961525e-16, -2.55871713e-17, -8.52182998e-16,
-8.87730619e-17, -1.91686944e-16, 1.165308049e-15, 2.77555756e-17,
2.47631776e-16, -3.33934269e-16, -1.90687272e-16, 7.31836466e-17,
3.94649591e-17, -1.83772268e-16, -7.14489232e-16, -3.04043978e-16,
2.61184303e-16, -3.65918233e-17, -1.36175793e-16, -5.32993788e-16,
-3.28923843e-17, 8.48579169e-01], 'test\\adriana lima (3).jpg': array([-1.31654274e+00, 2.39066362e+00, 2.83919661e+00, 9.70867245e+00,
1.18869617e-16, -2.6128013e-16, 6.93889390e-18, -3.81639165e-17,
2.24646690e-16, -9.7782998e-16, 1.59963189e-15, -4.31187294e-16,
-2.82759927e-16, -2.82759927e-16, 1.03885943e-15, 3.78169718e-16,
-6.28055222e-16, 7.15798727e-16, 1.71087183e-16, 6.32386797e-16,
-4.62526474e-16, -4.03323208e-17, -7.91467586e-18, 1.75287071e-16,
-2.08794242e-16, 1.19262239e-17, -7.01695646e-16, 2.93168267e-16,
7.63278329e-17, 7.28583866e-17, 1.73472348e-16, 2.28398881e-16,
-2.63244287e-16, 1.88651178e-16, -1.21438643e-17, 4.63388099e-16,
-3.18781229e-16, -2.82759927e-16, 1.03885943e-15, 3.78169718e-16,
-6.28055222e-16, 7.15798727e-16, 1.71087183e-16, 6.32386797e-16,
-4.62526474e-16, 9.29853964e-16, -2.88166817e-17, 1.29236899e-16,
-1.88848746e-16, 4.45173412e-16, 6.41140885e-16, 9.11163586e-16,
5.87515837e-16, 5.19846328e-01], 'test\\adriana lima (4).jpg': array([-2.08120479e-01, -1.43807362e-01, -4.17754637e-01, -1.23351613e+00,
1.38349348e+01, -5.37764278e-17, 2.76688394e-16, 4.65258257e-16,
-3.04043978e-16, -1.88463385e-17, -1.88463385e-16, 4.83973729e-16,
-3.05311322e-16, -1.00000000e+00, 2.76251485e-16, -2.04226310e-16,
1.69352322e-16, -3.33340228e-16, 3.04226310e-16, -3.641763510e-16,
5.87406617e-17, -2.46113893e-17, 4.06521608e-16, 1.58727198e-16,
7.82369288e-16, 8.06646461e-17, -2.91658384e-16, 8.06646461e-17,
6.29784622e-16, 3.07479736e-16, -3.1181655e-16, 4.23706299e-16,
3.27429056e-16, -4.11129464e-16, -3.39385031e-15, 2.91867225e-16,

```

Hasil Fungsi / Prosedur CalculateEigenFace

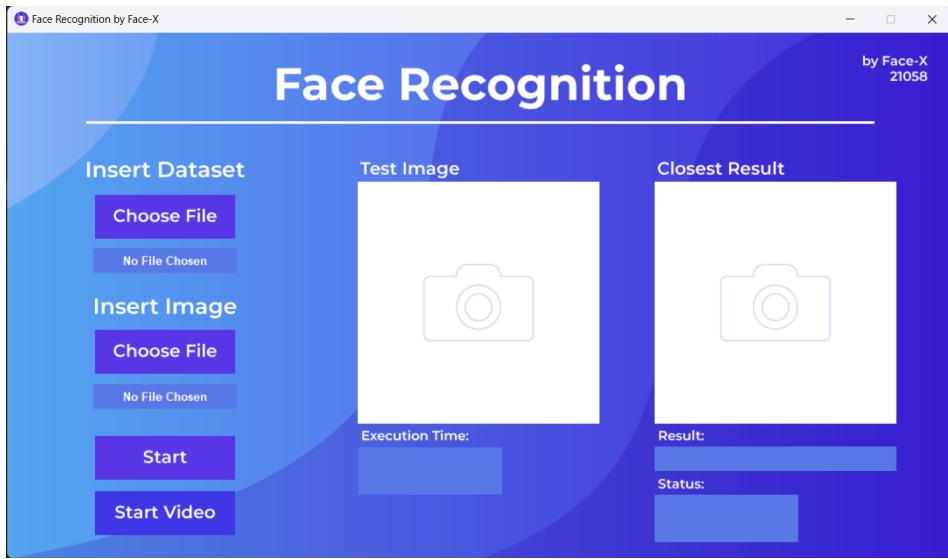
VI. Main

```

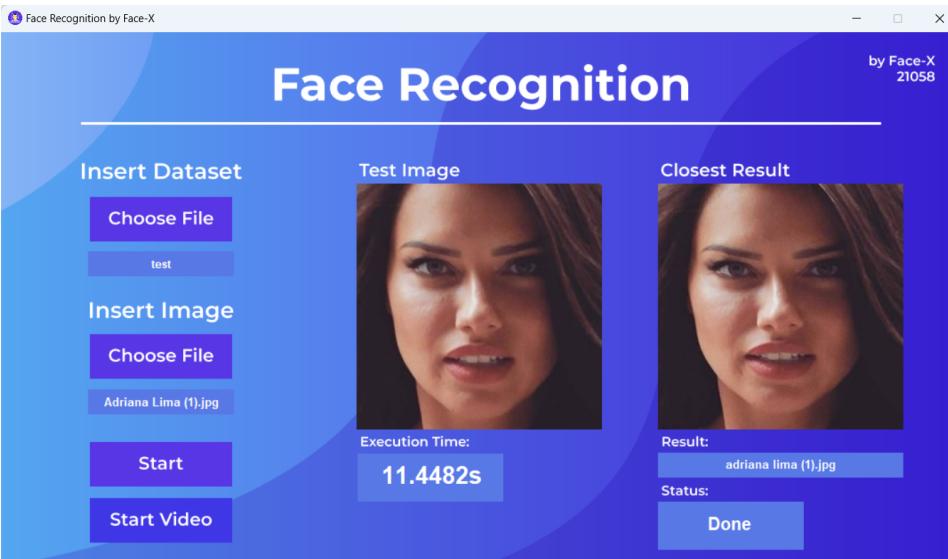
Updating Database...
Database berhasil dibuat (src/Database/database.pck)
Database selisih berhasil dibuat (src/Database/databaseSelisih.pck)
Database eigenface berhasil dibuat (src/Database/databaseEigenface.pck)
Result: test\\adriana lima (1).jpg
Distance: 0.0

```

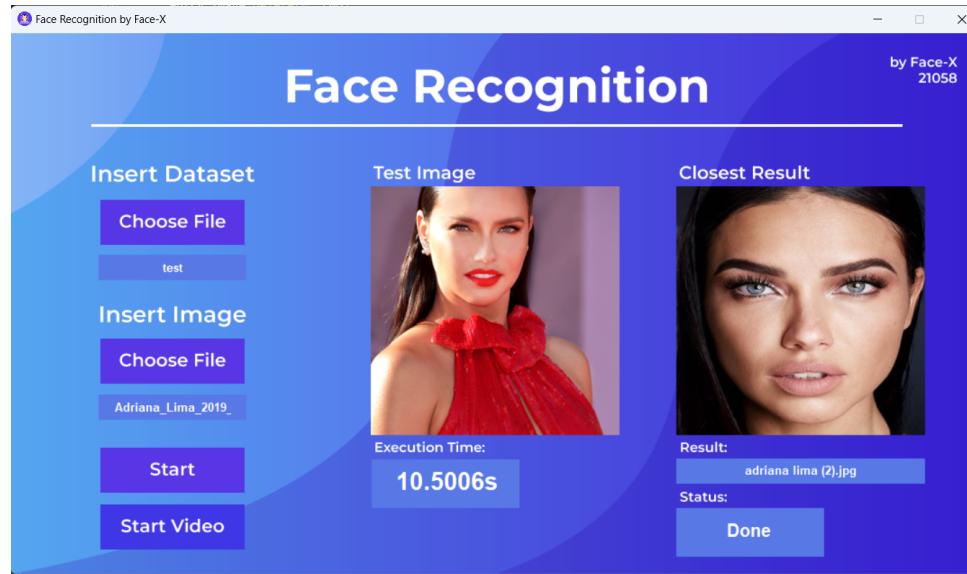
Deskripsi (Inisialisasi) : Terdapat 50 training image dalam subfolder test. Program dijalankan untuk pertama kali, dilakukan pembuatan database.



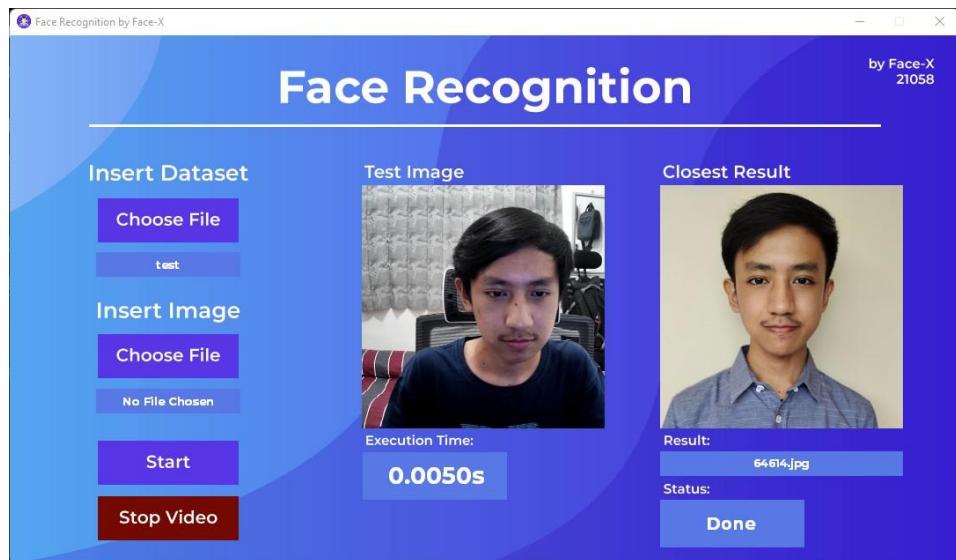
Deskripsi (Tampilan GUI) : Ketika program dijalankan, GUI dengan tampilan di atas akan muncul. Terdapat opsi insert dataset untuk memilih folder training image, insert image untuk menginput testface, start untuk memulai pemrosesan pengenalan wajah, start video untuk mengaktifkan video capture. Persegi kosong di sebelah kiri akan menampilkan input testface, persegi kosong di sebelah kanan akan menampilkan hasil wajah terdekat. Execution time menampilkan lama waktu eksekusi program mulai dari upload database hingga wajah terdekat ditemukan. Result akan menampilkan nama file dengan wajah terdekat. Status menampilkan keadaan program ketika processing atau perhitungan selesai.



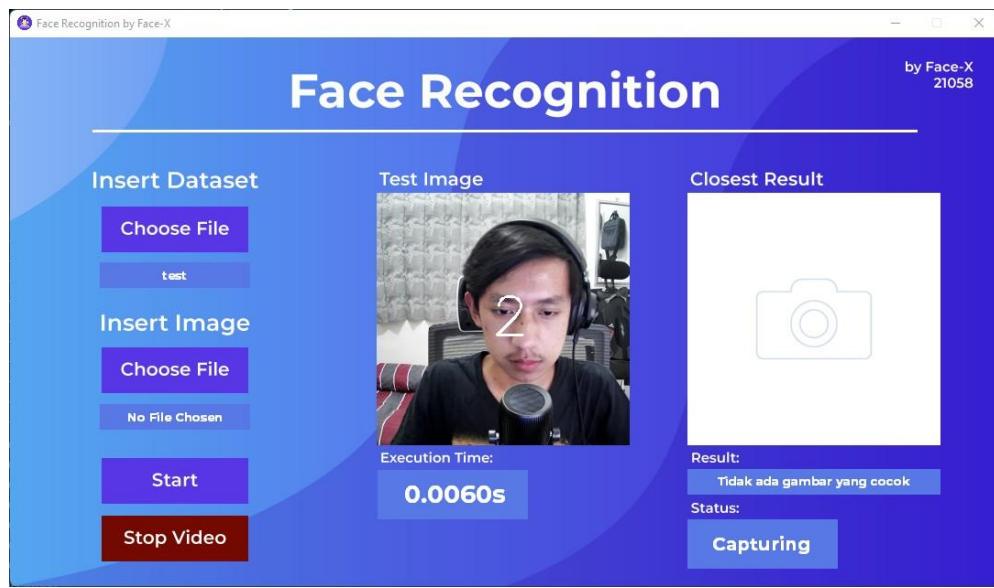
Deskripsi (Testface 1) : Input image yang dipilih adalah Adriana Lima(1).jpg. Ketika dijalankan, program menampilkan foto yang sama, adriana lima (1).jpg. Hal tersebut disebabkan foto testface yang diinput berasal dari folder training image.



Deskripsi (Testface 2): Input image adalah Adriana_Lima_2019_.jpg. Ketika dijalankan, program menampilkan foto adriana lima (2).jpg dari database. Disimpulkan program dapat mendeteksi kemiripan antara kedua wajah tersebut.



Deskripsi (Testface 3): Video capture dinyalakan sehingga GUI akan menampilkan rekaman kamera pada kotak kosong sebelah kiri. Ketika dijalankan, program menampilkan foto 64614.jpg. Disimpulkan program mendeteksi kemiripan antara kedua wajah tersebut.



Deskripsi (Testface 4): Video capture dinyalakan dan program dijalankan. Pada tes kali ini, foto 64614.jpg dihapus dari database sehingga euclidean distance terkecil antara input foto dengan tiap vektor dalam database melebihi batas maksimal. Karena melebihi batas maksimal, dianggap tidak terdapat foto yang cocok dengan foto input sehingga tidak ditampilkan foto apa pun pada kotak result.

BAB 5

Kesimpulan, Saran, dan Refleksi

I. Kesimpulan

Hasil yang dicapai melalui Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri:

- A. Program yang dapat digunakan dalam sistem pengenalan wajah (face recognition) dengan menggunakan Eigenface.
- B. Pengaplikasian konsep dasar matriks, nilai eigen, vektor eigen, dan Eigenface dalam sistem pengenalan wajah.
- C. Pengaplikasian bidang front-end dan back-end dalam pembuatan GUI dengan menggunakan Tkinter.
- D. Pengaplikasian OpenCV untuk image processing dan real.

II. Saran

Hal yang dapat dikembangkan berdasarkan hasil Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri:

- A. Efisiensi algoritma inisialisasi database. Sebelum program dijalankan, dibutuhkan waktu yang cukup lama untuk membuat database sehingga pemrosesan testface juga lama. Setelah database berhasil dibuat, pemrosesan testface selanjutnya tidak memerlukan waktu yang lama.
- B. Efisiensi algoritma perhitungan vektor eigen dan nilai eigen. Pada program digunakan metode QR untuk mencari vektor eigen dan nilai eigen. Algoritma looping masih belum efisien dan memiliki banyak perhitungan yang diulangi, padahal sebenarnya tidak diperlukan.

III. Refleksi

Pelajaran yang dapat diambil dari Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri:

- A. Pentingnya komunikasi antar anggota kelompok. Komunikasi yang efektif akan menghasilkan pekerjaan yang lebih maksimal, setiap anggota dapat memahami hasil pekerjaan anggota lainnya dengan lebih baik.
- B. Pentingnya manajemen waktu. Tugas besar perlu dicicil dan dikerjakan secepat mungkin, tidak menunda-nunda pengeroaan tugas besar agar tidak menumpuk dengan tugas-tugas lain.

Referensi

1. [Perkalian matriks - Wikipedia bahasa Indonesia, ensiklopedia bebas](#)
2. [Nilai Eigen dan Vektor Eigen \(Bagian 1\) \(itb.ac.id\)](#)
3. [Nilai Eigen dan Vektor Eigen \(Bagian 2\) \(itb.ac.id\)](#)
4. [GramSchmidt.pdf \(ucla.edu\)](#)
5. [A Face Recognition System Based on Eigenfaces Method \(sciedirectassets.com\)](#)
6. [IMPLEMENTASI PENGENALAN WAJAH DENGAN METODE EIGENFACE.pdf - Google Drive](#)
7. [Feature extraction and similar image search with OpenCV for newbies | by Andrey Nikishaev | Machine Learning World | Medium](#)
8. [ML | Face Recognition Using Eigenfaces \(PCA Algorithm\) - GeeksforGeeks](#)
9. [Pengenalan Wajah Menggunakan Algoritma Eigenface Dan Euclidean Distance - Neliti](#)
10. [Python GUI Programming With Tkinter – Real Python](#)
11. [Create UI using Tkinter in Python \(tutorialsteacher.com\)](#)
12. [How to build a GUI with PyQt - LogRocket Blog](#)

Repository github

<https://github.com/akmaldika/Algeo02-21058.git>

Video demo

<https://youtu.be/TuGA37E6Tl4>