

PROJECT BASED LEARNING 1

APLIKASI NILAI EIGEN DAN EIGEN FACE PADA PENGENALAN WAJAH



Dosen Pengampu: Drs. Bambang Harjito, M.App.Sc, PhD

Disusun Oleh:

- 1. AYU SANIATUS SHOLIHAH : L0124005**
- 2. FADHIL RUSADI : L0124013**
- 3. MUHAMAD NABIL FANNANI : L0124135**

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

DAFTAR ISI

BAB 1	1
DESKRIPSI MASALAH.....	1
1.1 Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Projek	2
1.4. Manfaat Projek	2
BAB 2	3
TEORI SINGKAT	3
2.1. Pengenalan Wajah (Face Recognition)	3
2.2. Perkalian Matriks	3
2.3. Nilai Eigen.....	4
2.4. Vektor Eigen.....	4
2.5. Eigenface	4
2.5. Euclidean Distance	5
BAB 3	6
3.1 Variabel Proyek.....	6
3.2 Jenis Proyek.....	6
3.3 Tahapan Pengembangan Sistem.....	6
3.3.1 Persiapan Dataset	7
3.3.2 Pra-pemrosesan Gambar	7
3.3.3 Proyeksi PCA dan Pembentukan Eigenface	7
3.3.4 Proses Pengenalan Wajah	8
3.3.5 Pembuatan Antarmuka	9
3.3.6 Pengujian Sistem.....	9
3.4 Teknik Pengumpulan Data	9
3.4.1. Visualisasi Komponen PCA (Eigenfaces)	9
3.4.2. Visualisasi Nilai Eigen (Norm Vektor Eigen) dan Threshold	10
3.4.3. Visualisasi Vektor Eigenface Pertama	10
3.4.4. Evaluasi Akurasi Model.....	10
3.5 Analisis Data	11
BAB 4	12
EKSPERIMEN.....	12

BAB 5	28
KESIMPULAN, SARAN, DAN REFLEKSI	28
DAFTAR PUSTAKA	30
LAMPIRAN	31

DAFTAR GAMBAR

Gambar 3. 1 Tahapan Pengembangan Sistem.....	7
Gambar 3. 2 Alur Algoritma Eigen Face	8
Gambar 4. 1 Tahapan Eksperimen	12
Gambar 4. 2 Eksperimen ke-1.....	13
Gambar 4. 3 Hasil eksperimen ke-1	13
Gambar 4. 4 Grafik Nilai Eigen dan Vektor Eigen.....	14
Gambar 4. 5 Plot Vektor Eigenface	15
Gambar 4. 6 Evaluasi Akurasi Model.....	15
Gambar 4. 7 Eksperimen ke-2.....	16
Gambar 4. 8 Hasil Eksperimen ke-2	16
Gambar 4. 9 Grafik Nilai Eigen dan Vektor Eigen.....	17
Gambar 4. 10 Plot Vektor Eigenface	17
Gambar 4. 11 Evaluasi Akurasi Model.....	18
Gambar 4. 12 Eksperimen ke-3.....	19
Gambar 4. 13 Hasil Eksperimen ke-3	19
Gambar 4. 14 Menambah Ambang Batas Threshold.....	20
Gambar 4. 15 Hasil Eksperimen ke-3	20
Gambar 4. 16 Gambar 22 Grafik Nilai Eigen dan Vektor Eigen.....	21
Gambar 4. 17 Plot Vektor Eigenface	21
Gambar 4. 18 Evaluasi Akurasi Model.....	22
Gambar 4. 19 Ekperimen ke-4	23
Gambar 4. 20 Hasil Eksperimen ke-5	23
Gambar 4. 21 Menaikkan Ambang Batas Threshold.....	23
Gambar 4. 22 Hasil Ekperimen ke-4.....	24
Gambar 4. 23 Grafik Nilai Eigen dan Vektor Eigen.....	24
Gambar 4. 24 Plot Vektor Eigenface	25
Gambar 4. 25 Evaluasi Akurasi Model.....	26

DAFTAR TABEL

Tabel 4. 1 Hasil Pengenalan Wajah	27
---	----

BAB 1

DESKRIPSI MASALAH

1.1 Latar Belakang

Pengenalan wajah (face recognition) merupakan salah satu teknologi biometrik yang digunakan untuk mengidentifikasi wajah seseorang berdasarkan citra digital. Teknologi ini memiliki peran penting dalam berbagai bidang, terutama pada sistem keamanan, seperti kontrol akses, pengawasan, dan verifikasi identitas. Dalam implementasinya, sistem pengenalan wajah bekerja dengan cara mempelajari bentuk wajah dari kumpulan citra yang disimpan dalam basis data, kemudian mencocokkannya dengan citra baru yang ingin diidentifikasi.

Terdapat berbagai metode yang dapat digunakan dalam sistem pengenalan wajah, seperti penggunaan Euclidean distance, cosine similarity, Local Binary Pattern Histogram (LBPH), Convolutional Neural Network (CNN), serta metode berbasis Principal Component Analysis (PCA) dan Eigenface. Metode LBPH cukup sederhana namun kurang tahan terhadap perubahan pose dan pencahayaan. Sementara itu, CNN memiliki akurasi tinggi, tetapi memerlukan data besar dan daya komputasi tinggi (Parkhi et al., 2015). Sebaliknya, metode PCA dan Eigenface menawarkan solusi yang efisien karena mampu mereduksi dimensi data dan mengekstraksi fitur utama dari citra wajah sehingga cocok untuk aplikasi berskala kecil hingga menengah (Turk & Pentland, 1991).

Proyek ini bertujuan membangun sistem pengenalan wajah berbasis Eigenface dan PCA menggunakan Python. Proses dimulai dengan tahap pelatihan, di mana kumpulan citra wajah dikonversi menjadi citra grayscale dan direpresentasikan dalam bentuk matriks. Dari representasi matriks tersebut, dihitung eigenvector utama untuk membentuk Eigenface. Selanjutnya, dalam tahap pencocokan, citra yang diuji akan dibandingkan dengan citra dalam database menggunakan jarak Euclidean untuk menentukan kemiripan. Jika nilai jarak minimum berada di bawah nilai ambang tertentu, maka citra dikenali; jika tidak, maka dianggap tidak dikenali.

Dalam proyek ini, fungsi ekstraksi fitur menggunakan pustaka seperti OpenCV atau PIL, namun perhitungan eigenvector diimplementasikan mandiri tanpa menggunakan fungsi bawaan. Dengan pendekatan ini, mahasiswa tidak hanya belajar menggunakan teknologi yang ada, tetapi juga memahami logika matematis di baliknya.

1.2. Rumusan Masalah

1. Bagaimana tingkat akurasi pengenalan wajah menggunakan algoritma Eigenface dan PCA dengan metode euclidean distance
2. Bagaimana menganalisis penggunaan algoritma Eigenface dan PCA dengan metode euclidean distance untuk pengenalan wajah manusia

1.3. Tujuan Proyek

1. Mengetahui tingkat akurasi penggunaan algoritma Eigenface dan PCA dengan metode euclidean distance
2. Mampu menganalisis penggunaan algoritma Eigenface dan PCA dengan metode euclidean distance untuk pengenalan wajah manusia

1.4. Manfaat Proyek

1. Memberikan pengalaman praktis dalam membangun sistem biometrik berbasis open source.
2. Menjadi dasar pengembangan sistem pengenalan wajah yang efisien dan terjangkau.

BAB 2

TEORI SINGKAT

2.1. Pengenalan Wajah (Face Recognition)

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

2.2. Perkalian Matriks

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

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

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

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

2.3. Nilai Eigen

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

2.4. Vektor Eigen

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

2.5. Eigenface

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

2.5. Euclidean Distance

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

Rumus jarak Euclidean antara dua vektor x dan y adalah

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

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

BAB 3

IMPLEMENTASI PROGRAM

Proyek ini dilaksanakan secara berkelompok oleh mahasiswa menggunakan laptop masing-masing pada bulan Mei 2025. Perangkat lunak yang digunakan dalam proyek ini meliputi Python, OpenCV, NumPy, dan Streamlit sebagai antarmuka berbasis Graphical User Interface (GUI), PIL, os dan pathlib. Dataset citra wajah yang digunakan diperoleh dari sumber publik di Kaggle yaitu TaylorSwift sebanyak 131 gambar dan TomCruise sebanyak 192 gambar dan digunakan untuk keperluan pelatihan serta pengujian sistem.

3.1 Variabel Proyek

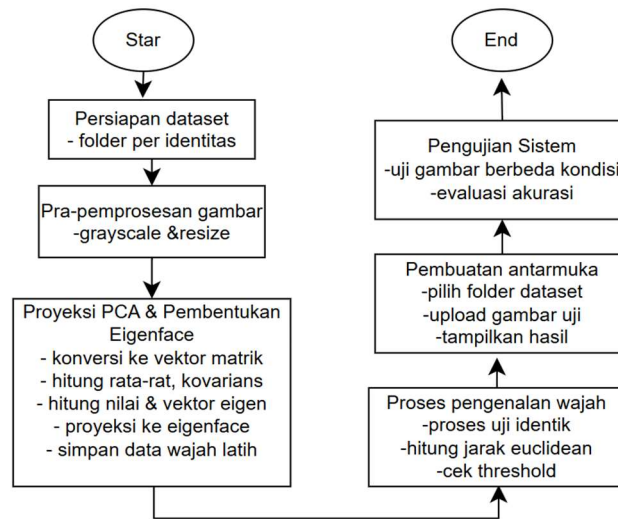
Pada proyek ini memiliki tiga variable diantaranya sebagai berikut:

1. Variabel bebas yang digunakan dalam proyek ini yaitu Path folder dataset, gambar input
2. Variabel terikat dalam proyek ini yaitu status pengenalan wajah (terkenali atau tidak) berdasarkan hasil perhitungan Euclidean Distance terhadap dataset PCA (eigenface).
3. Variabel kontrol dalam proyek ini yaitu ukuran gambar, format gambar, bilai ambang batas threshold Euclidean Distance yang digunakan (dapat disesuaikan, default pada 500), jumlah gambar pelatihan dalam folder dataset.

3.2 Jenis Proyek

Jenis proyek ini merupakan penelitian eksperimen kuantitatif berbasis rekayasa perangkat lunak. Penelitian dilakukan dengan membangun sistem pengenalan wajah menggunakan metode eigenface dan mengimplementasikan perhitungan nilai eigen, vektor eigen, serta jarak Euclidean

3.3 Tahapan Pengembangan Sistem



Gambar 3. 1 Tahapan Pengembangan Sistem

3.3.1 Persiapan Dataset

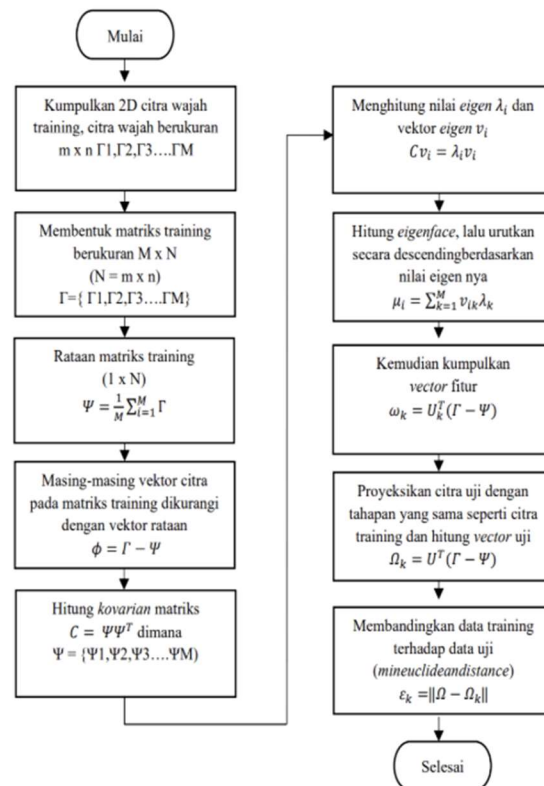
Tahap awal dalam pengembangan sistem pengenalan wajah adalah menyiapkan dataset citra wajah. Dataset citra wajah yang digunakan diperoleh dari sumber publik di Kaggle yaitu TaylorSwift sebanyak 131 gambar dan TomCruise sebanyak 192 gambar dan digunakan untuk keperluan pelatihan serta pengujian sistem. Yang nantinya akan digunakan sebagai data latih untuk membentuk model pengenalan. Setiap gambar wajah dikumpulkan dalam kondisi pencahayaan dan ekspresi yang relatif konsisten guna meningkatkan akurasi sistem.

3.3.2 Pra-pemrosesan Gambar

Setelah dataset siap, dilakukan pra-pemrosesan untuk mempersiapkan gambar agar dapat diproses lebih lanjut. Langkah ini meliputi konversi gambar ke format grayscale, normalisasi ukuran citra agar seragam, serta perataan intensitas cahaya untuk mengurangi pengaruh noise. Proses ini bertujuan agar data yang masuk ke tahap selanjutnya bersih dan memiliki kualitas seragam.

3.3.3 Proyeksi PCA dan Pembentukan Eigenface

Tahapan selanjutnya adalah menerapkan metode Principal Component Analysis (PCA) untuk melakukan reduksi dimensi terhadap citra wajah. PCA digunakan untuk mengekstraksi fitur utama dari gambar, menghasilkan representasi wajah dalam bentuk eigenface. Eigenface merupakan kombinasi linier dari wajah-wajah yang ada dalam dataset dan digunakan sebagai basis ruang fitur. Setiap gambar wajah kemudian diproyeksikan ke ruang eigenface dan diubah menjadi vektor ciri.



Gambar 3. 2 Alur Algoritma Eigen Face

3.3.4 Proses Pengenalan Wajah

Pada tahap ini, sistem melakukan pengenalan terhadap gambar uji dengan cara memproyeksikan gambar tersebut ke ruang eigenface dan membandingkannya dengan vektor ciri yang telah tersimpan dari data latih. Pengukuran kemiripan dilakukan

dengan menggunakan metode jarak, seperti Euclidean Distance, untuk menentukan identitas wajah yang paling mirip dengan citra uji.

3.3.5 Pembuatan Antarmuka

Untuk mempermudah pengguna dalam mengakses dan mengoperasikan sistem, dibuat antarmuka grafis yang sederhana dan interaktif. Antarmuka ini memungkinkan pengguna untuk melakukan input gambar, pelatihan model, serta melakukan pengujian dengan lebih mudah dan efisien, tanpa harus berinteraksi langsung dengan kode program.

3.3.6 Pengujian Sistem

Tahap akhir adalah melakukan pengujian terhadap sistem yang telah dibangun. Pengujian dilakukan menggunakan data uji yang berbeda dari data latih untuk mengevaluasi performa sistem dalam mengenali wajah. Hasil pengujian ini digunakan untuk mengukur tingkat akurasi, ketahanan terhadap variasi ekspresi dan pencahayaan, serta efektivitas keseluruhan dari metode yang digunakan.

3.4 Teknik Pengumpulan Data

Teknik pengumpulan data yang akan digunakan dalam proyek ini terdiri atas visualisasi komponen Principal Component Analysis (PCA), visualisasi nilai eigen, visualisasi vektor eigen, dan evaluasi akurasi model. Seluruh teknik pengumpulan data dilakukan di VS Code dengan bahasa pemrograman Python. Semua data diintegrasikan dalam aplikasi interaktif berbasis streamlit.

3.4.1. Visualisasi Komponen PCA (Eigenfaces)

Komponen utama hasil PCA disebut eigenfaces, yaitu vektor eigen dari matriks kovarians citra wajah yang sudah di-mean-center. PCA dilakukan menggunakan dekomposisi nilai singular (SVD):

$$X = U \Sigma V^T$$

3.4.2. Visualisasi Nilai Eigen (Norm Vektor Eigen) dan Threshold

Setiap vektor eigen memiliki nilai eigen yang diukur dengan **norm** vektornya:

$$\text{Norm}(v_i) = \|v_i\| = \sqrt{\sum_{j=1}^n v_{ij}^2}$$

Norm ini menunjukkan kontribusi vektor tersebut dalam representasi wajah.

Threshold untuk pengenalan wajah ditentukan berdasarkan jarak proyeksi wajah baru terhadap database:

$$\text{Threshold} = \alpha \max (\text{Jarak Euclidean proyeksi wajah})$$

3.4.3. Visualisasi Vektor Eigenface Pertama

Vektor eigen pertama divisualisasikan sebagai grafik satu dimensi yang menunjukkan bobot tiap piksel terhadap komponen utama. Vektor ini adalah bagian dari transformasi:

$$y = W^T(x - \mu)$$

di mana y adalah representasi wajah di ruang PCA, W matriks eigenfaces, X citra input, dan μ wajah rata-rata.

3.4.4. Evaluasi Akurasi Model

Evaluasi menggunakan metode Leave-One-Out Cross Validation (LOOCV).

Wajah diuji dengan menghitung jarak Euclidean proyeksi ke basis:

$$\text{Jarak} = \|y_{\text{input}} - y_{\text{basis}}\|$$

Jika jarak terkecil $<$ threshold, wajah dikenali; jika tidak, dianggap "Tidak Dikenal". Akurasi dihitung sebagai:

$$\text{Akurasi} = \left(\frac{\text{Jumlah prediksi benar}}{\text{Total gambat}} \right) \times 100\%$$

3.5 Analisis Data

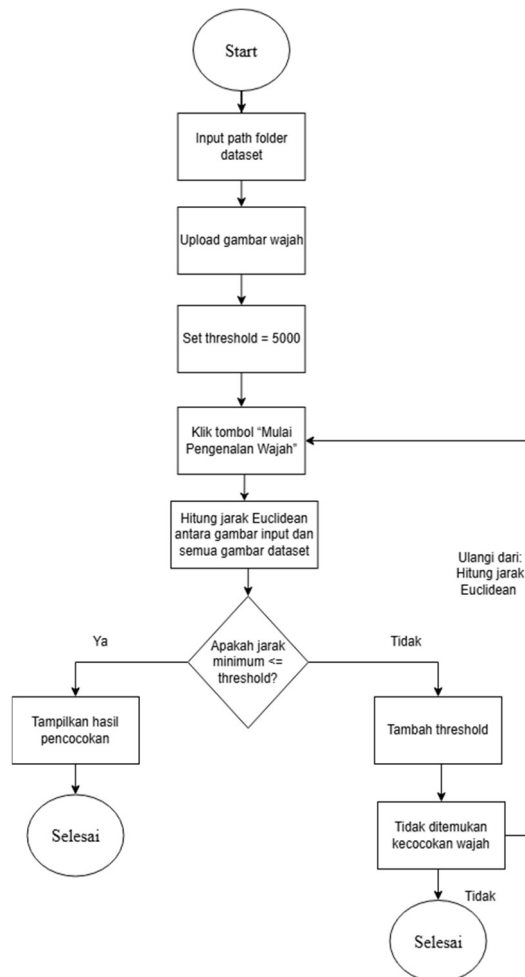
Analisis data dilakukan dengan pendekatan statistik deskriptif berbasis Principal Component Analysis (PCA). Proses meliputi perhitungan nilai eigen menggunakan normalisasi SVD, visualisasi komponen utama dan vektor eigen (eigenface), serta eksperimen variasi nilai threshold untuk membandingkan pengaruhnya terhadap akurasi pengenalan wajah. Evaluasi performa sistem dilakukan melalui confusion matrix dan perhitungan akurasi klasifikasi.

BAB 4

EKSPERIMEN

Dalam eksperimen ini, digunakan empat gambar uji yang terdiri dari dua gambar internal (berasal dari dataset pelatihan) dan dua gambar eksternal (tidak termasuk dalam dataset). Gambar internal dipilih dari masing-masing kelas, yaitu satu gambar wajah Tom Cruise dan satu gambar wajah Taylor Swift. Sementara itu, gambar eksternal merupakan citra wajah acak yang tidak terdapat dalam dataset, digunakan untuk menguji kemampuan sistem dalam menolak wajah yang tidak dikenal.

4.1 Tahapan Eksperimen



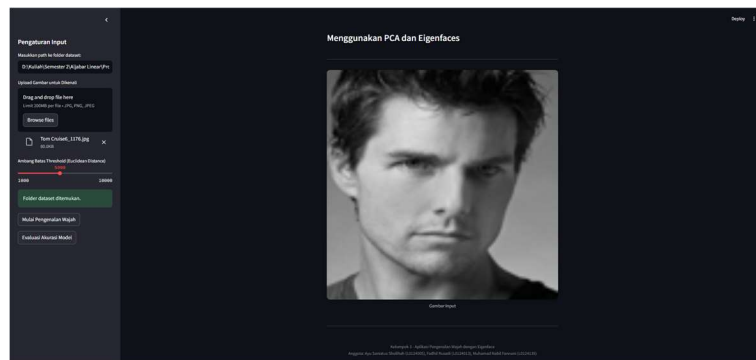
Gambar 4. 1 Tahapan Eksperimen

4.2 Analisis

Internal (berasal dari dataset pelatihan)

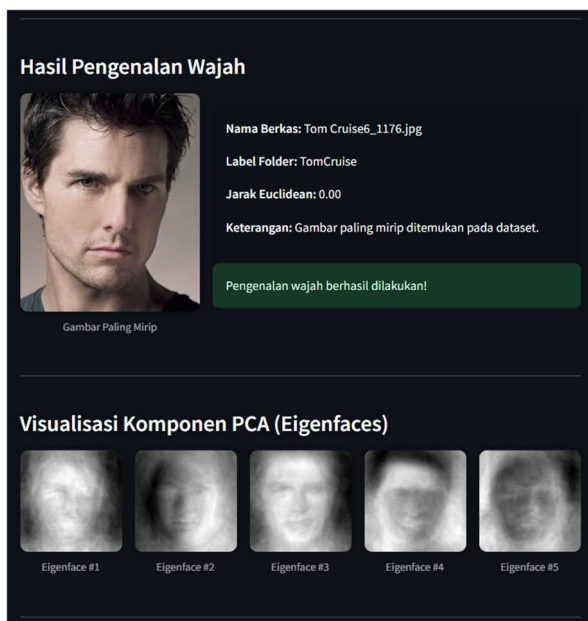
Tom Cruise

Diuji dengan Threshold 5000



Gambar 4. 2 Eksperimen ke-1

Hasil Pengenalan Wajah



Gambar 4. 3 Hasil eksperimen ke-1

Pengenalan wajah berhasil dilakukan!

Nama Berkas : Tom

Cruise6_1176.jpg

Label Folder : TomCruise

Jarak Euclidean : 0.00

Grafik Nilai Eigen dan Vektor Eigen

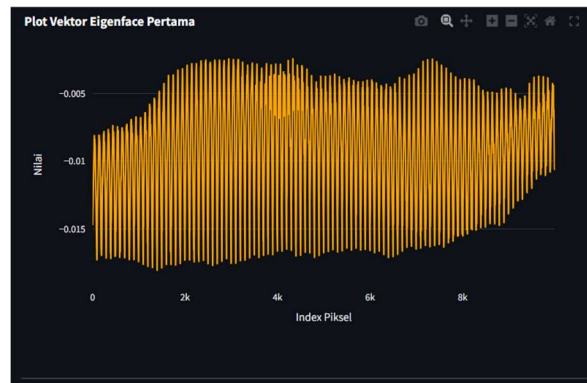


Gambar 4. 4 Grafik Nilai Eigen dan Vektor Eigen

Grafik "Distribusi Norm Vektor Eigenface" ini memvisualisasikan kontribusi setiap komponen eigen yang diperoleh dari analisis PCA, dengan sumbu horizontal menunjukkan "Komponen Eigen ke-" dan sumbu vertikal merepresentasikan "Norm Vektor". Pada grafik, terlihat bahwa semua nilai norm komponen eigenface (garis biru) sangat rendah dan mendekati nol, yang merupakan hasil dari proses normalisasi eigenfaces yang diterapkan dalam algoritma. Garis putus-putus merah pada nilai 5000 menandai threshold yang digunakan pada tahap pengenalan wajah sebagai kriteria jarak Euclidean, yang menentukan apakah suatu wajah input cukup mirip dengan wajah dalam dataset untuk dianggap dikenali.

Plot vektor Eigenface

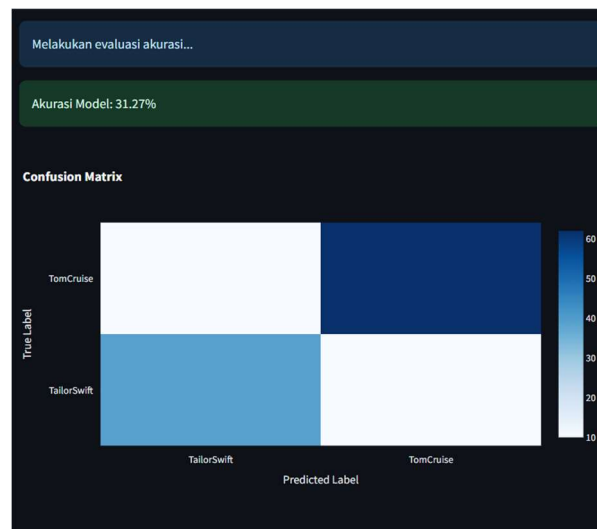
Grafik "Plot Vektor Eigenface Pertama" memvisualisasikan nilai-nilai (bobot) yang diberikan kepada setiap piksel dari citra wajah yang telah diratakan menjadi vektor satu dimensi (dengan indeks piksel dari 0 hingga 10.000). Sumbu vertikal menunjukkan nilai bobot yang berkisar antara sekitar -0.015 hingga -0.005. Pola yang terbentuk adalah serangkaian fluktuasi cepat dan kompleks yang membentuk gelombang, mengindikasikan bagaimana eigenface pertama menangkap variasi yang dominan dan seringkali halus antar piksel dalam dataset. Setiap titik pada grafik ini



Gambar 4. 5 Plot Vektor Eigenface

Menunjukkan kontribusi spesifik piksel tersebut terhadap pola variasi utama, di mana nilai-nilai yang lebih tinggi (mendekati -0.005) atau lebih rendah (mendekati -0.015) mengindikasikan area piksel yang sangat berperan dalam membedakan wajah satu sama lain dalam dataset.

Evaluasi Akurasi Model

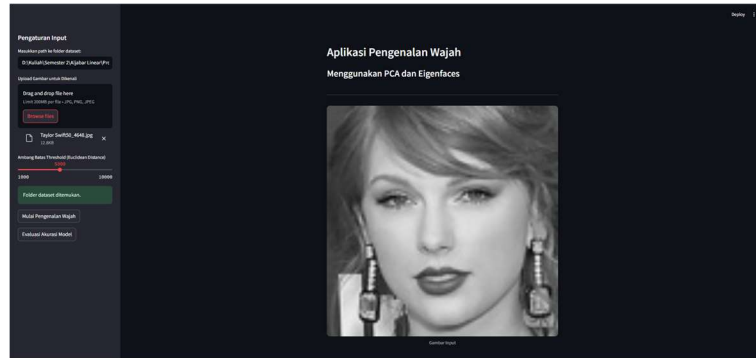


Gambar 4. 6 Evalusi Akurasi Model

Aplikasi dapat melakukan pengenalan wajah, kinerja klasifikasinya saat ini masih sangat terbatas, dengan akurasi hanya 31.27%. Confusion matrix secara jelas menguraikan bahwa sebagian besar kesalahan berasal dari model yang salah mengidentifikasi satu individu sebagai individu lainnya.

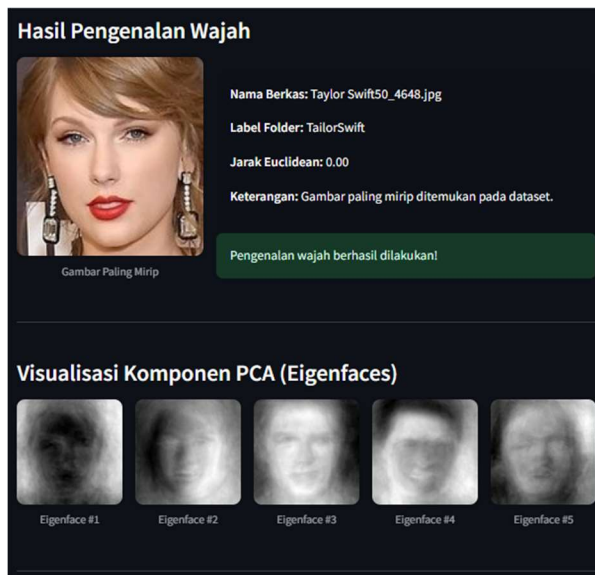
Taylor Swift

Diuji dengan Threshold 5000



Gambar 4. 7 Eksperimen ke-2

Hasil Pengenalan Wajah



Gambar 4. 8 Hasil Eksperimen ke-2

Pengenalan wajah berhasil dilakukan!

Nama Berkas : Taylor Swift50_4648.jpg

Label Folder : TaylorSwift

Jarak Euclidean : 0.00

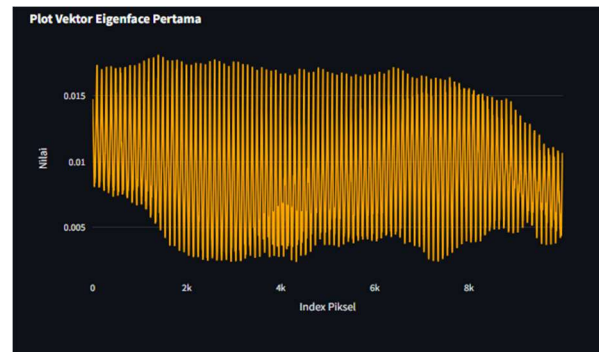
Grafik Nilai Eigen dan Vektor Eigen



Gambar 4. 9 Grafik Nilai Eigen dan Vektor Eigen

Grafik "Distribusi Norm Vektor Eigenface" ini memvisualisasikan kontribusi setiap komponen eigen yang diperoleh dari analisis PCA, dengan sumbu horizontal menunjukkan "Komponen Eigen ke-" dan sumbu vertikal merepresentasikan "Norm Vektor". Pada grafik, terlihat bahwa semua nilai norm komponen eigenface (garis biru) sangat rendah dan mendekati nol, yang merupakan hasil dari proses normalisasi eigenfaces yang diterapkan dalam algoritma. Garis putus-putus merah pada nilai 5000 menandai threshold yang digunakan pada tahap pengenalan wajah sebagai kriteria jarak Euclidean, yang menentukan apakah suatu wajah input cukup mirip dengan wajah dalam dataset untuk dianggap dikenali.

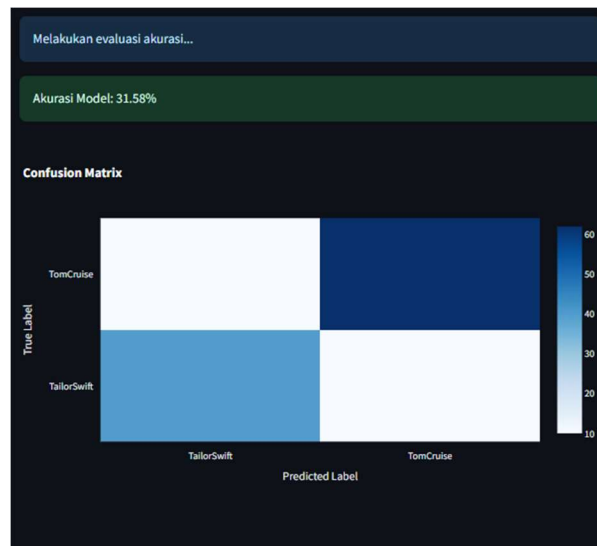
Plot Vektor Eigenface



Gambar 4. 10 Plot Vektor Eigenface

Grafik "Plot Vektor Eigenface Pertama" memvisualisasikan bobot atau nilai kontribusi setiap piksel dari citra wajah yang telah diratakan menjadi vektor satu dimensi (dengan indeks piksel dari 0 hingga 10.000). Sumbu vertikal menunjukkan nilai bobot yang bersifat positif, berkisar antara sekitar 0.005 hingga 0.015. Pola yang terbentuk adalah serangkaian fluktuasi cepat dan kompleks yang membentuk gelombang, mengindikasikan bagaimana eigenface pertama menangkap variasi yang dominan dan seringkali halus antar piksel dalam dataset. Setiap titik pada grafik ini menunjukkan kontribusi spesifik piksel tersebut terhadap pola variasi utama, di mana nilai-nilai yang lebih tinggi atau lebih rendah mengindikasikan area piksel yang sangat berperan dalam membedakan wajah satu sama lain dalam dataset.

Evaluasi Akurasi Model



Gambar 4. 11 Evaluasi Akurasi Model

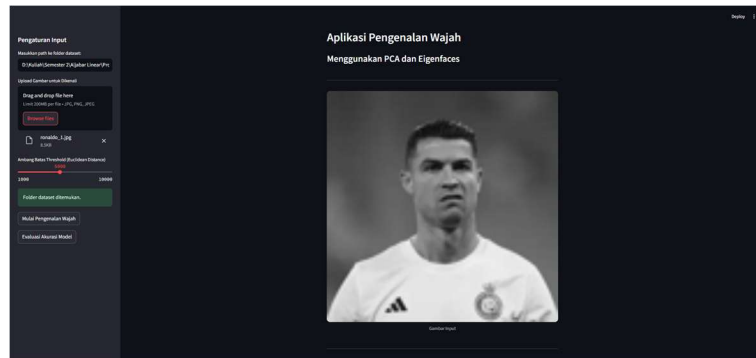
Aplikasi dapat melakukan pengenalan wajah, kinerja klasifikasinya saat ini masih sangat terbatas, dengan akurasi hanya 31.58%. Confusion matrix secara jelas

menguraikan bahwa sebagian besar kesalahan berasal dari model yang salah mengidentifikasi satu individu sebagai individu lainnya.

Eksternal Dataset (di luar dataset)

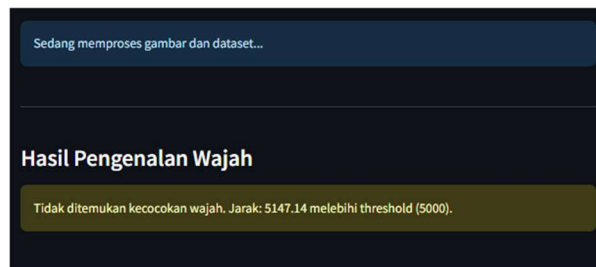
Cristiano Ronaldo

Diuji dengan Threshold 5000



Gambar 4. 12 Eksperimen ke-3

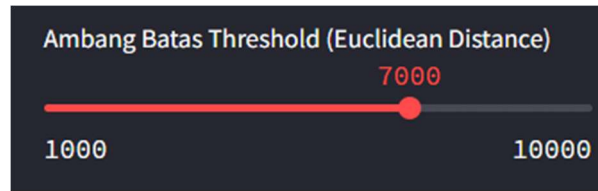
Hasil Pengenalan Wajah



Gambar 4. 13 Hasil Eksperimen ke-3

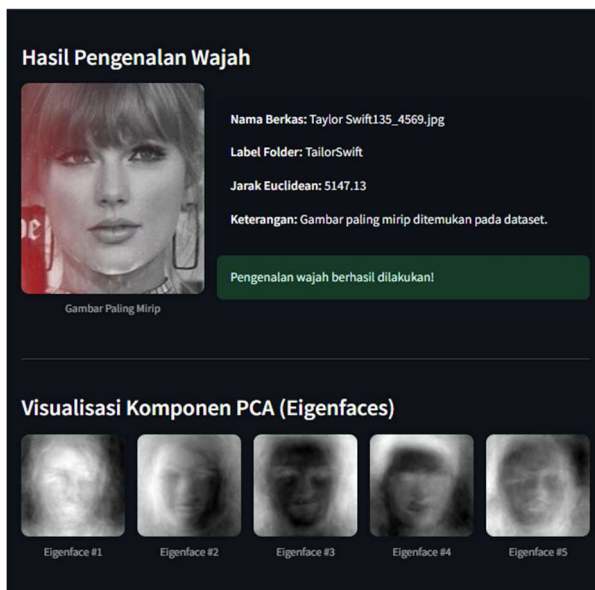
Tidak ditemukan kecocokan

Menambahkan threshold menjadi 7000



Gambar 4. 14Menambah Ambang Batas Threshold

Hasil Pengenalan Wajah



Pengenalan wajah berhasil dilakukan!

Nama Berkas : Taylor

Swift135_4569.jpg

Label Folder : TaylorSwift

Jarak Euclidean : 5147.13

Gambar 4. 15 Hasil Eksperimen ke-3

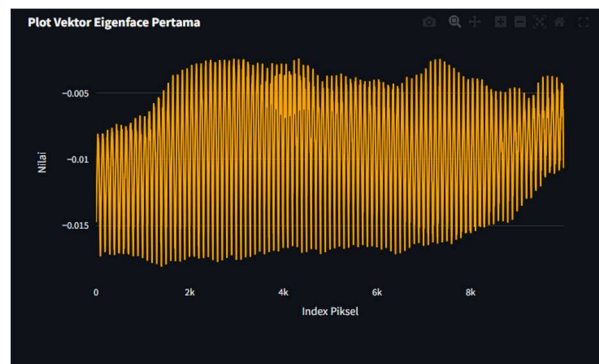
Grafik Nilai Eigen dan Vektor Eigen



Gambar 4. 16 Gambar 22 Grafik Nilai Eigen dan Vektor Eigen

Grafik "Distribusi Norm Vektor Eigenface" ini memvisualisasikan kontribusi setiap komponen eigen yang diperoleh dari analisis PCA, dengan sumbu horizontal menunjukkan "Komponen Eigen ke-" dan sumbu vertikal merepresentasikan "Norm Vektor". Pada grafik, terlihat bahwa semua nilai norm komponen eigenface (garis biru) sangat rendah dan mendekati nol, yang merupakan hasil dari proses normalisasi eigenfaces yang diterapkan dalam algoritma. Garis putus-putus merah pada nilai 7000 menandai threshold yang digunakan pada tahap pengenalan wajah sebagai kriteria jarak Euclidean, yang menentukan apakah suatu wajah input cukup mirip dengan wajah dalam dataset untuk dianggap dikenali.

Plot vektor Eigenface

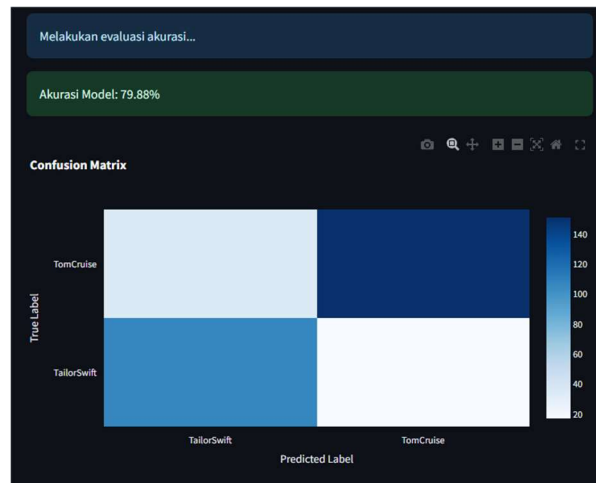


Gambar 4. 17 Plot Vektor Eigenface

Grafik "Plot Vektor Eigenface Pertama" memvisualisasikan bobot atau nilai kontribusi setiap piksel dari citra wajah yang telah diratakan menjadi vektor satu dimensi, dengan "Index Pksel" pada sumbu horizontal dan "Nilai" bobot pada sumbu vertikal, berkisar antara sekitar -0.015 hingga -0.005. Pola yang terbentuk adalah serangkaian fluktuasi cepat dan kompleks yang membentuk gelombang, mengindikasikan bagaimana eigenface pertama menangkap variasi yang dominan dan seringkali halus antar piksel dalam dataset. Setiap titik pada grafik ini menunjukkan kontribusi spesifik piksel tersebut terhadap pola variasi utama, di mana nilai-nilai yang

lebih tinggi atau lebih rendah mengindikasikan area piksel yang sangat berperan dalam membedakan wajah satu sama lain dalam dataset.

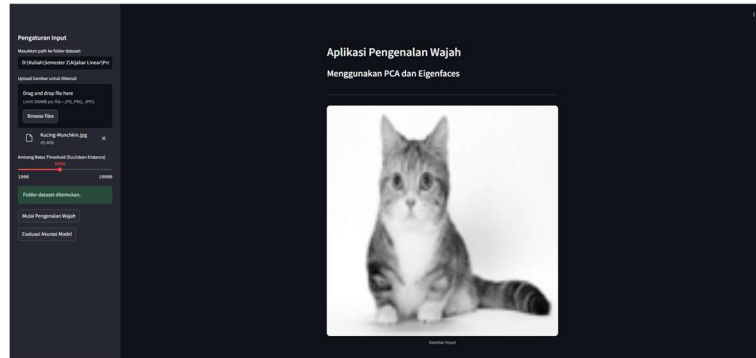
Evaluasi Akurasi Model



Gambar 4. 18 Evaluasi Akurasi Model

Matriks berikut menampilkan peningkatan signifikan pada "Akurasi Model" menjadi 79.88% selama evaluasi model pengenalan wajah. Meskipun akurasi tinggi ini dicapai, penting untuk dicatat bahwa peningkatan kinerja ini seringkali dapat dipengaruhi oleh penyesuaian parameter seperti nilai threshold. Pada beberapa kasus, threshold yang lebih tinggi (misalnya, 7000 dibandingkan 5000 sebelumnya) dapat menyebabkan model menemukan kecocokan yang "paling mirip" dalam dataset, bahkan ketika gambar input sebenarnya tidak identik atau tidak ada di dalam dataset pelatihan. Confusion Matrix yang terlampir merefleksikan perbaikan ini dengan menunjukkan dominasi prediksi benar (area terang pada diagonal utama) dan penurunan kesalahan klasifikasi yang signifikan (area gelap di luar diagonal), mengindikasikan bahwa model kini lebih sering mengidentifikasi individu dengan tepat berdasarkan tingkat kemiripan yang diizinkan oleh threshold yang disesuaikan.

Kucing



Gambar 4. 19 Ekperimen ke-4

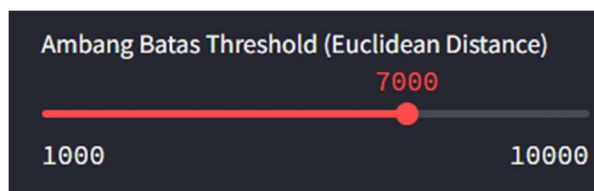
Hasil Pengenalan Wajah



Gambar 4. 20 Hasil Eksperimen ke-5

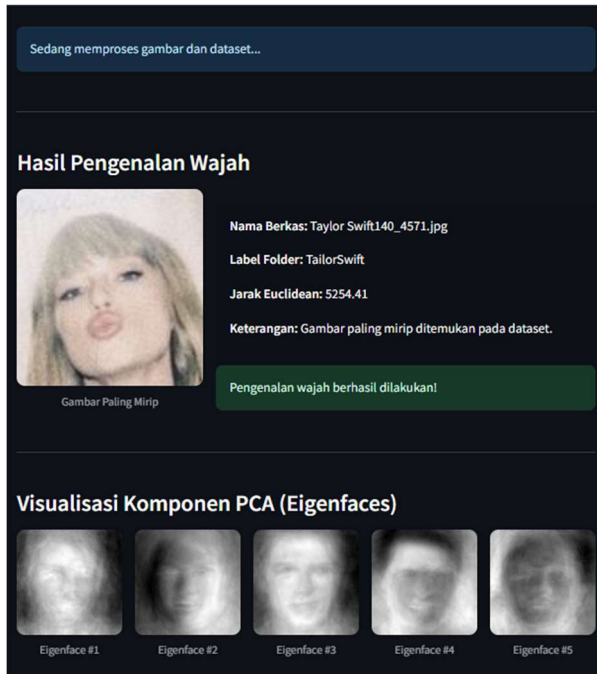
Tidak ditemukan kecocokan wajah. Jarak 5147.14 melebihi threshold (5000).

Menambahkan threshold menjadi 7000



Gambar 4. 21 Menaikkan Ambang Batas
Threshold

Hasil Pengenalan Wajah



Pengenalan wajah berhasil dilakukan!

Nama Berkas : Taylor

Swift140_4571.jpg

Label Folder : TaylorSwift

Jarak Euclidean : 5147.13

Gambar 4. 22 Hasil Ekperimen ke-4

Grafik Nilai Eigen dan Vektor Eigen

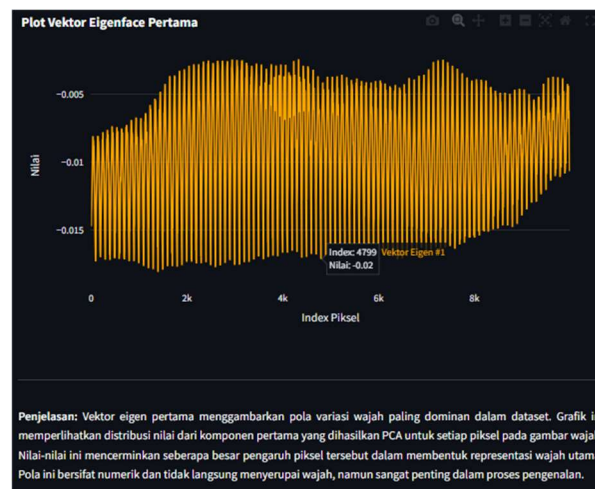


Gambar 4. 23 Grafik Nilai Eigen dan Vektor

Grafik "Distribusi Norm Vektor Eigenface" ini memvisualisasikan kontribusi setiap komponen eigen yang diperoleh dari analisis PCA, dengan sumbu horizontal

menunjukkan "Komponen Eigen ke-" dan sumbu vertikal merepresentasikan "Norm Vektor". Pada grafik, terlihat bahwa semua nilai norm komponen eigenface (garis biru) sangat rendah dan mendekati nol, yang merupakan hasil dari proses normalisasi eigenfaces yang diterapkan dalam algoritma. Garis putus-putus merah pada nilai 7000 menandai threshold yang digunakan pada tahap pengenalan wajah sebagai kriteria jarak Euclidean, yang menentukan apakah suatu wajah input cukup mirip dengan wajah dalam dataset untuk dianggap dikenali.

Plot vektor Eigenface

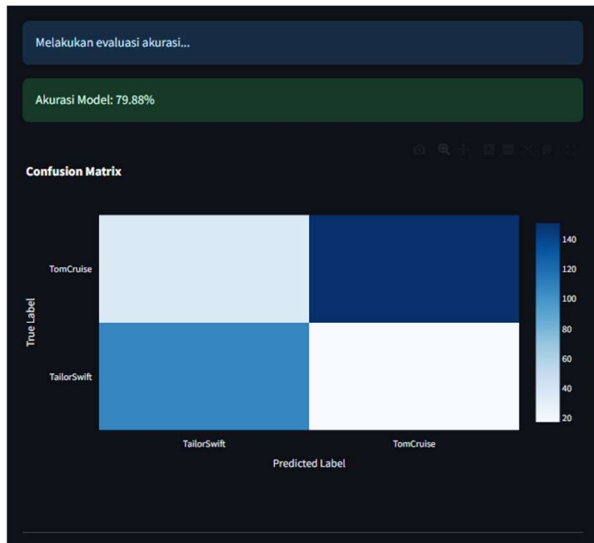


Gambar 4. 24 Plot Vektor Eigenface

Grafik "Plot Vektor Eigenface Pertama" memvisualisasikan bobot atau nilai kontribusi setiap piksel dari citra wajah yang telah diratakan menjadi vektor satu dimensi. Sumbu horizontal menunjukkan "Index Piksel", sementara sumbu vertikal merepresentasikan "Nilai" bobot, berkisar antara sekitar -0.015 hingga -0.005. Pola yang terbentuk adalah serangkaian fluktuasi cepat dan kompleks yang membentuk gelombang, mengindikasikan bagaimana eigenface pertama menangkap variasi yang dominan dan seringkali halus antar piksel dalam dataset. Setiap titik pada grafik ini menunjukkan kontribusi spesifik piksel tersebut terhadap pola variasi utama, di mana

nilai-nilai yang lebih tinggi atau lebih rendah mengindikasikan area piksel yang sangat berperan dalam membedakan wajah satu sama lain dalam dataset.

Evaluasi Akurasi Model



Gambar 4. 25 Evaluasi Akurasi Model

Menampilkan akurasi model pengenalan wajah yang signifikan sebesar 79.88%. Akurasi yang tinggi ini, meskipun menunjukkan kemampuan deteksi yang kuat, dapat dipengaruhi oleh penyesuaian parameter seperti nilai threshold. Dalam beberapa kasus, peningkatan threshold (misalnya dari 5000 menjadi 7000) memungkinkan model untuk mengidentifikasi kecocokan dengan "yang paling mirip" dari dataset, bahkan jika gambar input sebenarnya tidak identik atau bukan bagian dari dataset pelatihan asli. Confusion Matrix yang disertakan mengilustrasikan perbaikan kinerja ini dengan menunjukkan dominasi prediksi benar (area terang pada diagonal utama) dan penurunan signifikan dalam kesalahan klasifikasi (area gelap di luar diagonal), menegaskan bahwa model kini lebih efektif dalam mengidentifikasi individu berdasarkan kriteria kemiripan yang telah disesuaikan.

Tabel 4. 1 Hasil Pengenalan Wajah

Tes		Threshold	Jarak Euclidence	Dikenali/Tidak Dikenali	Dikenali Sebagai	Tingkat akurasi
Data Internal	Data Eksternal					
Tom Cruise	-	5000	0.00	Dikenali	Tom Cruise	31.27%
Taylor Swift	-	5000	0.00	Dikenali	Taylor Swift	31.58%
-	Cristiano Ronaldo	5000	5147.14	Tidak Dikenali	-	-
-	Cristiano	7000	5147.13	Dikenali	Taylor Swift	79.88%
-	Kucing	5000	5147.14	Tidak Dikenali	-	-
-	Kucing	7000	5147.13	Dikenali	Taylor Swift	79.99%

BAB 5

KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Berdasarkan hasil eksperimen pengujian pengenalan wajah menggunakan algoritma Eigenface berbasis PCA dan Euclidean Distance, aplikasi mampu mengenali wajah dari data internal secara sempurna dengan nilai jarak Euclidean 0.00 pada threshold 5000, sebagaimana ditunjukkan pada pengujian citra Tom Cruise dan Taylor Swift. Akurasi awal model dengan metode Leave-One-Out Cross Validation pada threshold yang sama mencapai 31%, dengan kesalahan klasifikasi dominan berupa identifikasi wajah yang keliru terhadap individu lain.

Penggunaan threshold memiliki peran krusial dalam kinerja sistem. Threshold 5000 menyebabkan data eksternal seperti citra Cristiano Ronaldo dan gambar kucing diklasifikasikan sebagai "Tidak Dikenali" akibat jarak Euclidean yang melebihi batas. Peningkatan threshold menjadi 7000 meningkatkan tingkat penerimaan sistem, memungkinkan citra eksternal dikenali, meskipun terkadang diklasifikasikan keliru sebagai "Taylor Swift". Peningkatan ini juga berdampak pada akurasi model yang meningkat signifikan menjadi 79,88% hingga 79,99%.

Visualisasi norm vektor eigenface menunjukkan nilai norm yang rendah akibat proses normalisasi dalam pembentukan eigenfaces, sedangkan vektor eigenface pertama menampilkan pola distribusi piksel yang kompleks dan menjadi dasar pembeda fitur wajah dalam proses klasifikasi.

5.2 Saran

1. Projek lebih lanjut diperlukan untuk menentukan nilai threshold yang optimal guna menyeimbangkan tingkat penerimaan dan akurasi klasifikasi, khususnya dalam menghadapi data eksternal.

2. Pengembangan dapat dilakukan dengan meningkatkan kualitas dan variasi dataset, seperti menambahkan gambar dengan pose, ekspresi, dan pencahayaan berbeda agar model lebih robust dalam mengenali wajah.
3. Evaluasi metode pengenalan wajah lain seperti LDA atau CNN perlu dilakukan untuk membandingkan performa dengan Eigenface, serta mengembangkan strategi klasifikasi wajah "Tidak Dikenali" yang lebih akurat dan adaptif.

5.3 Refleksi

Selama pengerjaan proyek "Aplikasi Nilai Eigen dan Eigenface pada Pengenalan Wajah", kami mendapatkan pengalaman praktis yang berharga dalam membangun sistem biometrik berbasis open source. Proyek ini memberikan pemahaman mendalam tentang logika matematis di balik algoritma Eigenface dan PCA, termasuk konsep perkalian matriks, nilai eigen, dan vektor eigen, karena kami mengimplementasikan perhitungan tersebut secara mandiri tanpa menggunakan fungsi bawaan. Tantangan utama yang dihadapi adalah mengoptimalkan performa model agar mampu mengenali wajah secara akurat, terutama dalam mengatasi variasi yang tidak terdapat dalam dataset pelatihan. Pengujian dengan data eksternal dan eksperimen dengan berbagai threshold mengajarkan pentingnya penyesuaian parameter dan evaluasi yang sistematis. Pengalaman ini secara signifikan meningkatkan pemahaman kami di bidang pengenalan wajah dan keterampilan pemrograman Python, serta menjadi dasar yang kuat untuk pengembangan sistem serupa di masa depan.

DAFTAR PUSTAKA

- Munir, R. (2022). Nilai Eigen dan Vektor Eigen (Bagian 1). Retrieved Mei 25, 2025 from <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf>
- Parkhi, O. M. (2015). Deep face recognition. In Proceedings of the British Machine Vision Conference (BMVC). From <https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf>
- Saefullah., A. (2015). Aplikasi Face Recognition Menggunakan Metode Eigen Face dan Principal Component Analysis.
- Saputra, W. M. (2013). Pengenalan wajah menggunakan algoritma Eigenface dan Euclidean Distance. *Journal of Informatics and Technology*, 2(1), 102–110. Retrieved Mei 24, 2025 from <http://ejournal-s1.undip.ac.id/index.php/joint>
- Strang, G. (2016). *Introduction to Linear Algebra*. Press., Wellesley-Cambridge(5th ed.).
- Turk, M. &. (1991). *Eigenfaces for recognition*. The Media Laboratory, Massachusetts Institute of Technology. Vision and Modeling Group.

LAMPIRAN

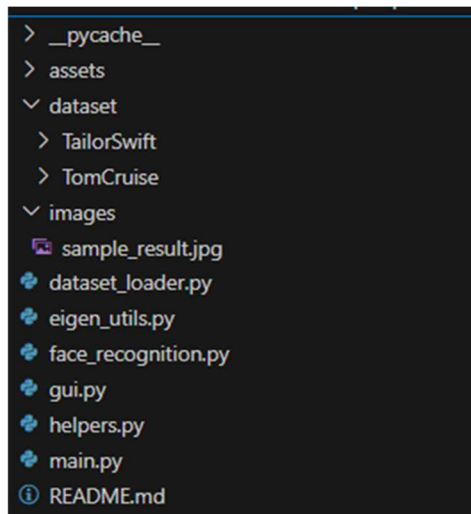
Link Repository Github:

<https://github.com/AyuSaniatusSholihah/APLIKASI-NILAI-EIGEN-DAN-EIGEN-FACE-PADA-PENGENALAN-WAJAH>

Link Demo:

<https://youtu.be/vqpjz5sQZ5w>

Struktur Folder:



1. Dataset_loader.py

```
import numpy as np
from PIL import Image

def load_dataset(dataset_path, image_size=(100, 100)):
    images = []
    labels = []
    filenames = []

    for root, _, files in os.walk(dataset_path):
        for file in files:
            if file.lower().endswith(('jpg', 'jpeg', 'png')):
                file_path = os.path.join(root, file)
                try:
                    img = Image.open(file_path).convert('L')
                    img = img.resize(image_size)
                    img_data = np.asarray(img, dtype=np.float64).flatten()

                    images.append(img_data)
                    labels.append(os.path.basename(root))
                    filenames.append(file)
                except Exception as e:
                    print(f"Gagal memproses {file_path}: {e}")

    return np.array(images), labels, filenames
```

2. eigen_utils.py

```

utils.py > h_eigenvekt
import numpy as np

def meancenter(images):
    mean_face = np.mean(images, axis=0)
    centered = images - mean_face
    return centered, mean_face

def h_matrikscov(data):
    return np.dot(data, data.T)

def pow_iter(A, num_iter=1000, tol=1e-6):
    b_k = np.random.rand(A.shape[1])
    b_k /= np.linalg.norm(b_k)

    for _ in range(num_iter):
        b_k1 = np.dot(A, b_k)
        b_k1_norm = np.linalg.norm(b_k1)
        if b_k1_norm == 0:
            break
        b_k1 /= b_k1_norm
        if np.linalg.norm(b_k - b_k1) < tol:
            break
        b_k = b_k1

    eigenvalue = np.dot(b_k.T, np.dot(A, b_k))
    return eigenvalue, b_k

def h_eigenvekt(data, num_components=None):
    cov_matrix = h_matrikscov(data)

    eigenvalues = []
    eigenvectors = []

    B = cov_matrix.copy()

    n_components = num_components if num_components else data.shape[0]

    for _ in range(n_components):
        val, vec = pow_iter(B)
        eigenvalues.append(val)
        eigenvectors.append(vec)

        B = B - val * np.outer(vec, vec)

    eigenvalues = np.array(eigenvalues)
    eigenvectors = np.array(eigenvectors).T

    eigenfaces = np.dot(data.T, eigenvectors)

    for i in range(eigenfaces.shape[1]):
        eigenfaces[:, i] /= np.linalg.norm(eigenfaces[:, i])

    return eigenfaces, eigenvalues

```

3. face_recognition.py

```
import numpy as np
from eigen_utils import meancenter, h_eigenvekt

def recogface(input_image, dataset_images, labels, filenames, threshold=5000, return_pca=False):
    centered_data, mean_face = meancenter(dataset_images)
    eigenfaces, _ = h_eigenvekt(centered_data)

    projections = np.dot(centered_data, eigenfaces)

    input_centered = input_image - mean_face
    input_projection = np.dot(input_centered, eigenfaces)

    distances = [np.linalg.norm(projection - input_projection) for projection in projections]
    min_index = np.argmin(distances)
    min_distance = distances[min_index]

    result = (filenames[min_index], labels[min_index], min_distance) if min_distance < threshold else (None, None, min_distance)

    if return_pca:
        return *result, eigenfaces, mean_face
    else:
        return result
```

4. helper.py

```
import numpy as np
from PIL import Image
import io

def read_upimg(uploaded_file, image_size=(100, 100)):
    try:
        image = Image.open(uploaded_file).convert('L')
        image = image.resize(image_size)
        image_vector = np.asarray(image, dtype=np.float64).flatten()
        return image, image_vector
    except Exception as e:
        print(f"Gagal membaca gambar: {e}")
        return None, None
```