



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem dilakukan dengan merealisasikan rancangan ke dalam bentuk aplikasi berbasis *website* yang bertujuan untuk mempermudah proses absensi tamu di Dinas Kearsipan Provinsi Sumatera Selatan. Sistem ini memanfaatkan teknologi *Face Recognition* dengan metode *Linear Discriminant Analysis (LDA)* untuk ekstraksi fitur dan *Euclidean Distance* untuk klasifikasi wajah. Aplikasi dibangun menggunakan HTML, CSS, dan *Bootstrap* untuk tampilan antarmuka yang responsif dan user-friendly, sedangkan backend dikembangkan menggunakan *PHP native* dan *Python* sebagai pendukung pengolahan data dan proses pengenalan wajah. Sistem ini dirancang agar dapat diakses langsung melalui browser perangkat komputer maupun mobile tanpa memerlukan instalasi tambahan atau konfigurasi rumit.

Tamu akan diarahkan untuk mengakses halaman absensi melalui *QR Code* yang tersedia di area penerimaan, lalu mengisi data kunjungan dan melakukan swafoto sebagai verifikasi identitas. Foto tersebut diproses oleh sistem melalui ekstraksi fitur dan klasifikasi wajah untuk memastikan bahwa identitas yang diklaim sesuai dengan data latih yang telah tersimpan sebelumnya dalam sistem. Semua data, baik isian formulir maupun citra wajah, secara otomatis tersimpan ke dalam basis data *MySQL* yang telah terintegrasi. Dengan penerapan sistem ini, proses absensi menjadi lebih efisien, akurat, cepat, dan aman dibandingkan dengan metode manual yang sebelumnya digunakan, serta mengurangi potensi kesalahan input dan manipulasi data.

4.2 Tampilan Antarmuka Sistem

Tampilan antarmuka pada sistem *E-Visitor* dirancang secara sederhana dan responsif agar dapat diakses baik melalui perangkat komputer maupun *mobile*. Antarmuka ini terdiri dari dua sisi, yaitu tampilan untuk admin dan untuk tamu.

Pada sisi admin, sistem menyediakan halaman, *log in*, *dashboard*, daftar tamu, kalender kunjungan, dan laporan kunjungan. Sementara pada sisi tamu, sistem



menampilkan halaman absensi yang dapat diakses melalui pemindaian *QR Code*, di mana tamu akan mengisi data kunjungan dan melakukan swafoto.

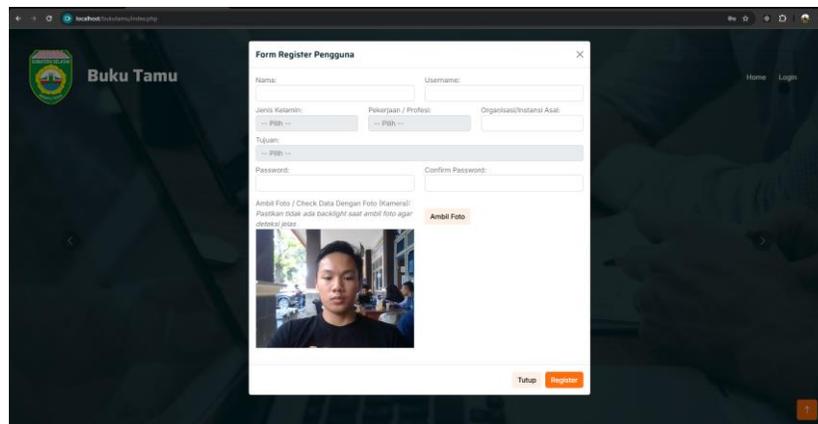
Berikut ini merupakan beberapa tampilan antarmuka dari sistem yang telah diimplementasikan :

4.2.1 *Landing Page*



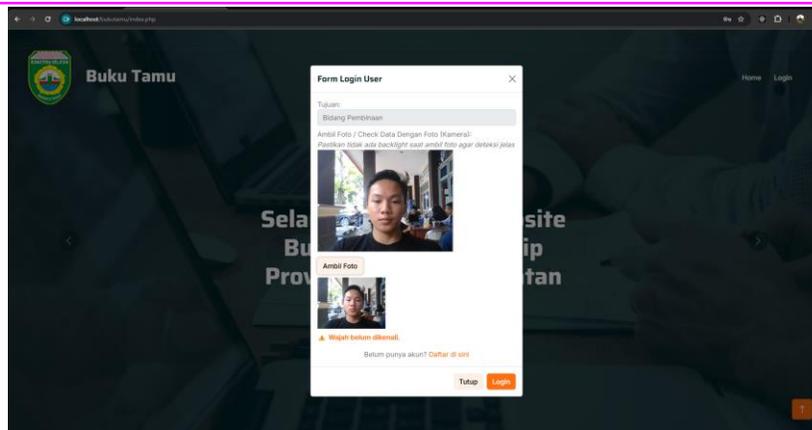
Gambar 4. 1 *Landing Page*

4.2.2 *Registrasi Tamu*



Gambar 4. 2 *Registrasi Tamu*

4.2.3 *Login Tamu*



Gambar 4.3 Login Tamu

4.2.4 Logout Tamu



Gambar 4.4 Logout Tamu

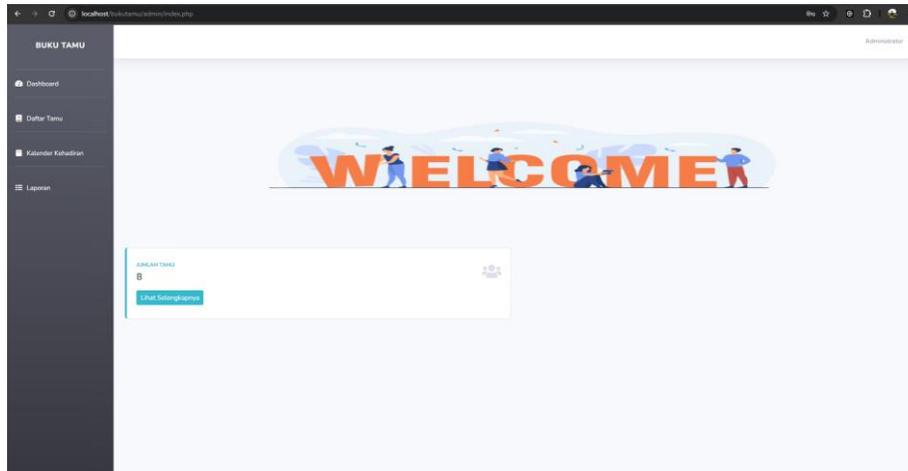
4.2.5 Login Admin





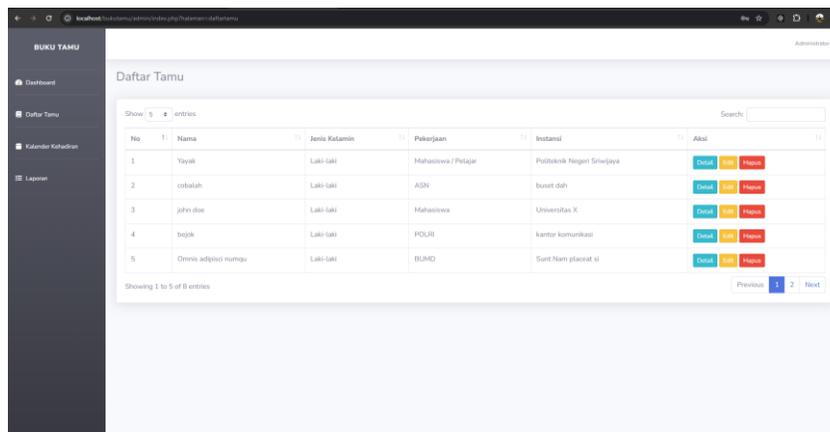
Gambar 4.5 Login Admin

4.2.6 Dashboard Admin



Gambar 4.6 Dashboard Admin

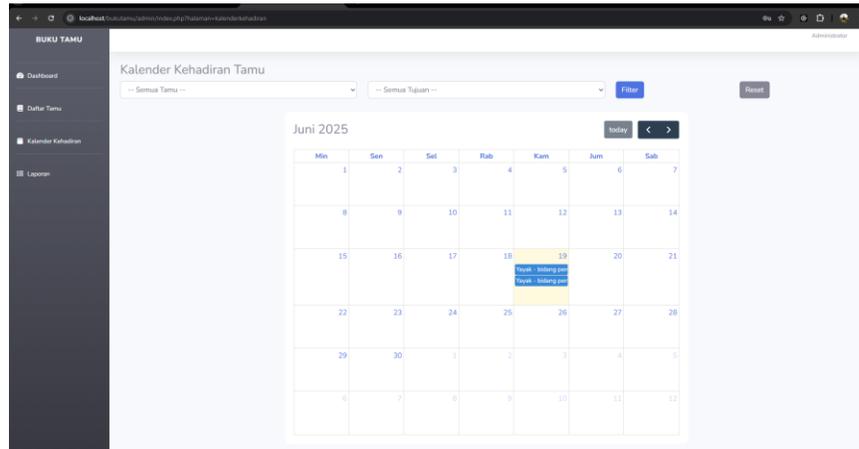
4.2.7 Daftar Tamu



Gambar 4.7 Daftar Tamu

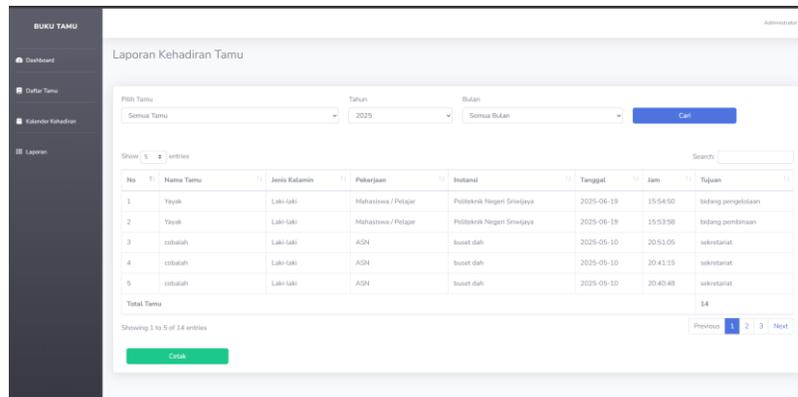


4.2.8 Kalender Tamu



Gambar 4. 8 Kalender Tamu

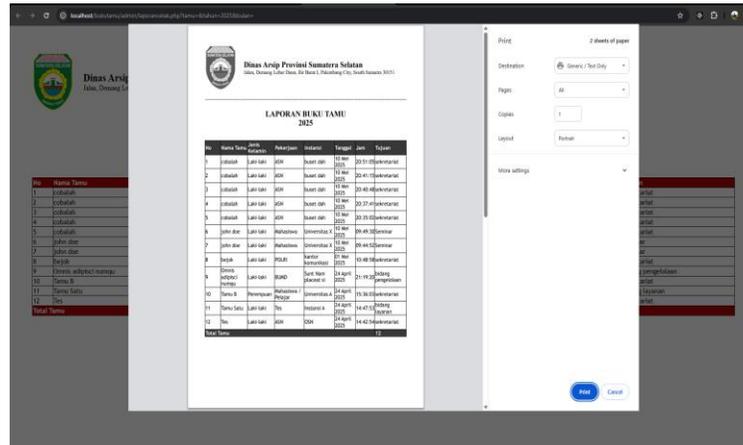
4.2.9 Laporan



Gambar 4. 9 Laporan



4.2.10 Cetak Laporan



Gambar 4. 10 Cetak Laporan

4.2.11 Edit Profil Admin



Gambar 4. 11 Edit Profil Admin

4.3 Pengujian Sistem

Pengujian sistem merupakan tahapan penting dalam proses pengembangan aplikasi untuk memastikan bahwa seluruh komponen berjalan sesuai dengan kebutuhan dan perencanaan yang telah dirancang sebelumnya. Dalam aplikasi *E-Visitor* ini, pengujian dilakukan secara menyeluruh untuk memastikan bahwa sistem dapat bekerja dengan baik, baik dari sisi teknis maupun dari sisi pengalaman pengguna (*user experience*). Tujuan utama dari pengujian ini adalah untuk menilai apakah fitur-fitur yang tersedia dalam sistem dapat dijalankan dengan lancar, memberikan hasil yang akurat, serta mampu memenuhi kebutuhan proses absensi tamu secara cepat, efisien, dan modern.



Pengujian sistem mencakup dua aspek utama, yaitu pengujian fungsional dan pengujian proses pengenalan wajah yang menggunakan metode *Linear Discriminant Analysis (LDA)* sebagai ekstraksi fitur dan *Euclidean Distance* sebagai metode klasifikasi. Kedua pengujian ini dijalankan secara paralel untuk memastikan integrasi antara antarmuka pengguna dan proses pengolahan data berjalan optimal.

4.3.1 Pengujian Fungsional

Pengujian fungsional berfokus pada pengujian setiap fitur dan alur kerja dalam sistem untuk memastikan bahwa seluruh fungsi berjalan sebagaimana mestinya sesuai dengan kebutuhan pengguna dan rancangan awal. Proses ini dilakukan dengan menjalankan satu per satu fitur yang tersedia, baik dari sisi pengguna tamu maupun dari sisi admin. Pada sisi admin, fitur yang diuji meliputi *Login admin*, *Dashboard*, daftar tamu, kalender kunjungan, laporan kunjungan, serta fitur cetak laporan. Sedangkan pada sisi tamu, pengujian dilakukan pada proses pemindaian *QR Code*, registrasi, pengisian data kunjungan, pengambilan swafoto, dan *Login*. Selama proses pengujian, setiap interaksi dicatat dan dievaluasi apakah sistem merespons dengan benar, serta apakah data dapat diproses dan disimpan dengan baik. Hasil pengujian ini menjadi indikator awal bahwa sistem telah stabil dan siap digunakan untuk pengujian lebih lanjut pada aspek pengenalan wajah.

Tabel 4. 1 Pengujian Fungsional

No	Fitur Yang Diuji	Status
1.	<i>Landing Page</i>	Berhasil
2.	Registrasi Tamu	Berhasil
3.	<i>Login Tamu</i>	Berhasil
4.	<i>Logout Tamu</i>	Berhasil
5.	<i>Login Admin</i>	Berhasil
6.	<i>Dashboard Admin</i>	Berhasil
7.	Daftar Tamu	Berhasil
8.	Kalender Tamu	Berhasil
9.	Laporan Kunjungan	Berhasil



10.	Cetak Laporan	Berhasil
11.	Edit Profil Admin	Berhasil

Berdasarkan hasil pengujian tersebut, seluruh fitur utama dalam sistem dapat berjalan dengan baik sesuai fungsinya.

4.3.2 Pengujian *Face Recognition*

Pengujian *Face Recognition* dilakukan untuk mengetahui tingkat keberhasilan sistem dalam mengenali wajah tamu berdasarkan data yang telah disimpan sebelumnya. Dalam pengujian ini, digunakan 30 orang partisipan, di mana masing-masing memiliki 5 citra wajah dengan ekspresi dan pose yang berbeda-beda. Dari 5 citra tersebut, 1 foto digunakan sebagai data latih dan 4 foto digunakan sebagai data uji.

4.3.3 Proses Pengujian Sistem

Pengujian dilakukan secara langsung terhadap 30 partisipan yang masing-masing melakukan :

- 1 kali registrasi (foto sebagai data latih)
- 4 kali *login*/absensi (foto sebagai data uji)

Total data uji sebanyak 120 citra wajah ($30 \text{ partisipan} \times 4 \text{ login}$). Data latih dan uji yang diperoleh diekstraksi menggunakan metode *LDA*, lalu diklasifikasikan menggunakan *Euclidean Distance* untuk menentukan apakah wajah berhasil dikenali atau tidak.

Tahapan perhitungan menggunakan *LDA* dijelaskan sebagai berikut:

1. Normalisasi vektor secara manual dilakukan untuk menyamakan skala data vektor fitur citra wajah. Ini penting karena nilai asli piksel dari gambar bisa sangat bervariasi, sehingga diperlukan penyesuaian agar setiap data memiliki kontribusi yang seimbang dalam proses perhitungan. Dengan menormalisasi vektor, sistem menjadi lebih stabil dan akurat dalam menghitung jarak atau perbedaan antar citra.

Rumus normalisasi vektor :

$$v_{norm} = \frac{v}{\|v\|}$$

Keterangan :



v : Vektor asli dari data

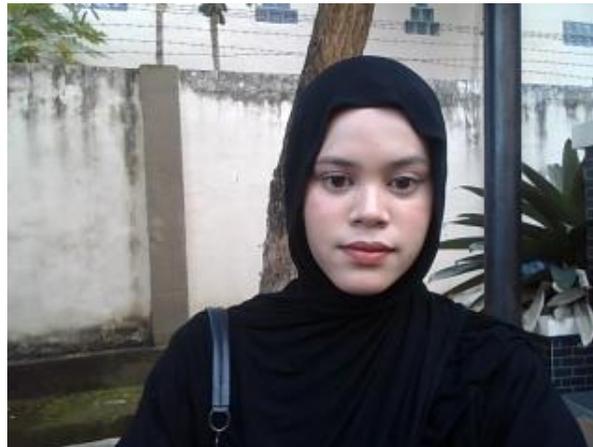
$\|v\|$: Panjang total dari vektor

v_{norm} : Vektor baru hasil normalisasi

```
11
12 def normalize(v):
13     return v / np.linalg.norm(v) if np.linalg.norm(v) != 0 else v
14
```

Gambar 4. 12 Kode Normalisasi Vektor

Diketahui :



Gambar 4. 13 Foto Partisipan yang dijadikan Data Vektor

$$v = [184, 226, 254, 190, 189, 190]$$

Ditanya :

Bagaimana bentuk vektor normalisasi dari data tersebut?

Penyelesaian :

$$v_1^2 = 184 = 184^2 = 33,856$$

$$v_2^2 = 226 = 226^2 = 51,076$$

$$v_3^2 = 254 = 254^2 = 64,516$$

$$v_4^2 = 190 = 190^2 = 36,100$$

$$v_5^2 = 189 = 189^2 = 35,721$$

$$v_6^2 = 190 = 190^2 = 36,100$$

$$\text{Total} = 33,856 + 51,076 + 64,516 + 36,100 + 35,721 + 36,100 = 257,369$$



$$\| v \| = \sqrt{257,369} = 507,3155$$

$$v_1 = \frac{184}{507,3155} = 0.3267$$

$$v_2 = \frac{226}{507,3155} = 0.4455$$

$$v_3 = \frac{254}{507,3155} = 0.5007$$

$$v_4 = \frac{190}{507,3155} = 0.3745$$

$$v_5 = \frac{189}{507,3155} = 0.3725$$

$$v_6 = \frac{190}{507,3155} = 0.3745$$

$$v_{norm} = [0.3267, 0.4455, 0.5007, 0.3745, 0.3725, 0.3745]$$

2. *Mean* per Kelas (μ_i) bertujuan untuk memperoleh nilai rata-rata dari setiap kelas data wajah. Nilai rata-rata ini mencerminkan pusat atau titik representatif dari fitur wajah dalam satu kelas, yang nantinya digunakan untuk membandingkan antar kelas dalam tahap pembelajaran *LDA*.

Rumus *Mean* per Kelas :

$$\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$$

```
102     if 0 in y_train and 1 in y_train:
103         class1 = X_train_lda[y_train == 0][:, 0]
104         class2 = X_train_lda[y_train == 1][:, 0]
105         mean1 = np.mean(class1)
106         mean2 = np.mean(class2)
```

Gambar 4. 14 Kode Penghitungan Mean per Kelas

Keterangan :

- μ_i : Mean untuk kelas ke-i
- x_{ij} : Nilai ke-j dari kelas ke-i
- n_i : Jumlah data dalam kelas ke-i



Σ : Notasi penjumlahan semua nilai

Diketahui :

Kelas 1 : [0.3627, 0.3745, 0.3725]

Kelas 2 : [0.4454, 0.5007, 0.3745]

Ditanya : Berapa nilai rata-rata (*mean*) untuk masing-masing kelas?

Penyelesaian :

Kelas 1 :

$$\mu_1 \frac{0.3627 + 0.3745 + 0.3725}{3} = \frac{1.1097}{3} = 0.3699$$

Kelas 2 :

$$\mu_2 \frac{0.4454 + 0.5007 + 0.3745}{3} = \frac{1.3206}{3} = 0.4402$$

3. *Mean Global* (μ) digunakan untuk menghitung rata-rata dari seluruh data wajah yang berasal dari semua kelas. Nilai ini menjadi acuan utama dalam menghitung variasi antar kelas (S_b), karena menunjukkan pusat distribusi keseluruhan data.

Rumus *Mean* per Global :

$$\mu = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^{n_i} x_{ij}$$

```
107 | mean_global = np.mean(np.concatenate((class1, class2)))
```

Gambar 4. 15 Kode Penghitungan Mean Global

Keterangan :

μ : Mean global untuk data semua kelas

x_{ij} : Nilai ke-j dari kelas ke-i

n_i : Jumlah data dalam kelas ke-i

Σ : Notasi penjumlahan semua nilai

N : Total seluruh data dari semua kelas

K : Jumlah kelas



Diketahui :

N : [0.3627, 0.3745, 0.3725, 0.4454, 0.5007, 0.3745], N = 6

Ditanya : Berapa rata-rata global (mean semua data)?

Penyelesaian :

$$\mu = \frac{0.3627 + 0.3745 + 0.3725 + 0.4454 + 0.5007 + 0.3745}{6}$$

$$\mu = \frac{2.4303}{6}$$

$$\mu = 0.4050$$

4. Matriks Sebaran Antar Kelas S_b (*Between-Class Scatter Matrix*) bertujuan untuk mengetahui sejauh mana perbedaan antar kelas. Semakin besar nilai S_b , maka semakin baik data tiap kelas dapat dipisahkan dari kelas lainnya, yang menjadi indikator penting dalam efektivitas pemisahan ruang pada metode *LDA*.

Rumus Matriks Sebaran Antar Kelas (S_b) :

$$S_b = \sum_{i=1}^c n_i (\mu_i - \mu)^2$$

```
109 | sb = len(class1) * (mean1 - mean_global) ** 2 + len(class2) * (mean2 - mean_global) ** 2
```

Gambar 4. 16 Kode Perhitungan Matriks Sebaran dalam Kelas

Keterangan :

C : Jumlah kelas

n_i : Jumlah data di kelas ke-i

μ_i : Mean kelas ke-i

μ : Mean global

Diketahui :

$$\mu_1 = 0.3699$$

$$\mu_2 = 0.4402$$



$$\mu = 0.4050$$

$$n_1 = 3$$

$$n_2 = 3$$

Ditanya : Berapa nilai sebaran antar kelas?

Penyelesaian :

Untuk kelas 1 :

$$\begin{aligned} n_1(\mu_1 - \mu)^2 &= 3 \times (0.3699 - 0.4051)^2 = 3 \times (-0.0352)^2 = 3 \times 0.00123904 \\ &= 0.00371712 \end{aligned}$$

Untuk kelas 2 :

$$\begin{aligned} n_2(\mu_2 - \mu)^2 &= 3 \times (0.4402 - 0.4051)^2 = 3 \times (0.0351)^2 = 3 \times 0.00123201 \\ &= 0.00369603 \end{aligned}$$

Total Sb :

$$Sb = 0.00371712 + 0.00369603 = 0.00741315$$

5. Matriks Sebaran Dalam Kelas S_w (*Within-Class Scatter Matrix*) dihitung untuk mengukur seberapa besar penyebaran atau variasi data dalam setiap kelas. Semakin kecil nilai S_w , maka data dalam satu kelas dianggap lebih konsisten, yang baik untuk pemisahan antar kelas karena data tidak terlalu menyebar.

Rumus Matriks Sebaran Dalam Kelas (S_w) :

$$S_w = \sum_{i=1}^c \sum_{x \in \text{kelas}-i} (x - \mu_i)^2$$

```
108 | sw = np.sum((class1 - mean1) ** 2) + np.sum((class2 - mean2) ** 2)
```

Gambar 4. 17 Kode Perhitungan Matriks Sebaran Dalam Kelas

Keterangan :

S_w : Within-class scatter (sebaran dalam kelas), ukuran variabilitas data dalam setiap kelas

$\sum_{i=1}^c$: Penjumlahan dari semua kelas, dari kelas ke-1 sampai kelas ke-C

$x \in \text{kelas} - i$: Artinya semua data yang ada di dalam kelas ke-i



x : Data individual

μ_i : Mean dari kelas ke-i

$(x - \mu_i)^2$: Selisih kuadrat antara data xxx dan rata-rata kelasnya

Diketahui :

Kelas 1 :

$$x_1 = (0.3627 - 0.3699)^2 = (-0.0072)^2 = 0.00005184$$

$$x_2 = (0.3745 - 0.3699)^2 = (0.0046)^2 = 0.00002116$$

$$x_3 = (0.3725 - 0.3699)^2 = (0.0026)^2 = 0.00000676$$

$$Sw_1 = 0.00005184 + 0.00002116 + 0.00000676 = 0.00007976$$

Kelas 2 :

$$x_1 = (0.4454 - 0.4402)^2 = 0.0052^2 = 0.00002704$$

$$x_2 = (0.5007 - 0.4402)^2 = (0.0605)^2 = 0.00366025$$

$$x_3 = (0.3745 - 0.4402)^2 = (-0.0657)^2 = 0.00431649$$

$$Sw_2 = 0.00002704 + 0.00366025 + 0.00431649 = 0.00800378$$

Ditanya : Berapa nilai sebaran dalam kelas?

Penyelesaian :

Total Sw :

$$Sw = Sw_1 + Sw_2 = 0.00007976 + 0.00800378 = 0.00808354$$

6. Proyeksi Ruang Baru (W) digunakan untuk menghitung rasio sebaran antar kelas terhadap dalam kelas. Hasil dari proses ini menentukan arah terbaik yang memisahkan data wajah antar kelas. Semakin tinggi nilainya, semakin baik kemampuan pemisahan fitur wajah dalam ruang baru tersebut.

Rumus Proyeksi Ruang Baru :

$$w = \frac{S_b}{S_w}$$

Keterangan :

W : Rasio antar kelas terhadap dalam kelas. Ini yang jadi arah pemisah data terbaik

S_b : Matriks sebaran antar kelas



S_w : Matriks sebaran dalam kelas

```
110 | | w = sb / sw
```

Gambar 4. 18 Kode Perhitungan Proyeksi Ruang Baru

Diketahui :

$$S_b = 0.00741313$$

$$S_w = 0.00808354$$

Ditanya : Berapa nilai rasio antara sebaran antar kelas dengan dalam kelas?

Penyelesaian :

$$w = \frac{0.00741313}{0.00808354}$$
$$w = 0.917188241$$

7. Arah proyeksi menunjukkan vektor arah terbaik yang dihasilkan dari perbedaan rata-rata antar kelas. Nilai ini penting untuk menentukan bagaimana sistem akan memproyeksikan data ke ruang *LDA* agar pemisahan antar kelas menjadi optimal.

Rumus Arah Proyeksi :

$$w = \frac{\mu_1 - \mu_2}{S_w}$$

```
111 | | w_proj = (mean1 - mean2) / sw
```

Gambar 4. 19 Kode Perhitungan Arah Proyeksi

Keterangan :

μ_1 : Mean dari kelas 1 setekah normalisasi

μ_2 : Mean dari kelas setekah normalisasi

S_w : Matriks sebaran dalam kelas

Diketahui :

$$\mu_1 = 0.3699$$

$$\mu_2 = 0.4402$$

$$S_w = 0.00808354$$

Ditanya : Berapa arah proyeksi optimal *LDA*?



Penyelesaian :

$$\mu_1 - \mu_2 = 0.3699 - 0.4402 = -0.0703$$

$$w = \frac{-0.0703}{0.00808354}$$

$$w = -8.6967$$

4.3.4 Klasifikasi dengan Euclidean Distance

Tahap akhir dari pengujian adalah menghitung jarak antara data uji dan data latih menggunakan metode *Euclidean Distance*. Jika jarak ini berada di bawah ambang batas (*threshold*), maka wajah dianggap dikenali. Metode ini efektif untuk membandingkan kemiripan antar dua vektor fitur wajah.

Rumus Klasifikasi Euclidean :

$$d(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

```

48 def predict_with_threshold(X_test, X_train, y_train, threshold=THRESHOLD):
49     dists = np.linalg.norm(X_train - X_test[0], axis=1)
50     min_idx = np.argmin(dists)
51     min_dist = dists[min_idx]
52     predicted_label = y_train[min_idx] if min_dist < threshold else None
53     return predicted_label, min_dist

```

Gambar 4. 20 Kode Perhitungan *Euclidean Distance*

Keterangan	:
A	: Vektor dari data latih
B	: Vektor data uji
a_i	: Elemen ke-i dari vector data latih
b_i	: Elemen ke-I dari vector data uji
$(a_i - b_i)^2$: Kuadrat selisih nilai fitur pada indeks ke-i
$\sum_{i=1}^n (a_i - b_i)^2$: Penjumlahan semua kuadrat selisih dari indeks 1 sampai n
$d(A, B)$: Nilai akhir jarak Euclidean antara dua vektor
n	: Jumlah total elemen dalam vektor
Diketahui	:



Data Latih A = [175,215,248,100,21,21]

Data Uji B = [184,226,254,190,189,190]

Ditanya : Berapa jarak Euclidean dari kedua vektor?

Tabel 4. 2 Tabel Vektor Fitur: Perbandingan antara Data Latih dan Data Uji

Index	Data Latih a_i	Data uji b_i	Selisih ($a_i - b_i$)	Kuadrat Selisih ($a_i - b_i$) ²
1	175	184	-9	81
2	215	226	-11	121
3	248	254	-6	36
4	100	190	-90	8100
5	21	189	-168	28224
6	21	190	-169	28561
$\sum_{i=1}^n (a_i - b_i)^2 = 9.070.061,3456$				

$$d(A, B) = \sqrt{9.070.061,3456} = 3011.16$$

```
PS C:\xampp\htdocs\phpldabukutamu> python recognizer.py uji_astri_1.jpg
Vektor Fitur Gambar Uji: [184 226 254 ... 190 189 190]
{"distance": 3011.16, "label": "astri", "recognized": true}
Vektor Fitur Gambar Uji: [184 226 254 ... 190 189 190]
{"distance": 3011.16, "label": "astri", "recognized": true}

=== PEMBUKTIAN MANUAL LDA ===

[Manual LDA] Mean Kelas 1 (mu1): 0.3699
[Manual LDA] Mean Kelas 2 (mu2): 0.4402
[Manual LDA] Mean Global (mu): 0.405
[Manual LDA] Sw = 0.00808354
[Manual LDA] Sb = 0.00741313
[Manual LDA] W (Sb / Sw) = 0.9171
[Manual LDA] W Proyeksi = (mu1 - mu2)/Sw = -8.6967
[Manual LDA] Proyeksi Data Uji (x=0.4100): -3.5656
```

Gambar 4. 21 Pembuktian Perhitungan Manual dengan Python

Berikut visualisasi data wajah dua orang partisipan:



Gambar 4. 22 Sample Data Citra Wajah Dua Orang dengan 5 Ekspresi yang Berbeda-Beda (2 dari 30 Orang)

Dari hasil uji awal, sistem mampu mengenali wajah masing-masing tamu dengan hasil sebagai berikut :

Tabel 4. 3 Hasil Pengujian *Face Recognition* Menggunakan *LDA* dan *Euclidean Distance*

No	Nama Partisipan	Jumlah Data Uji	Jumlah Data Latih	Data Dikenali Benar	Akurasi
1.	Alfarizi	4 foto	1 foto	4 foto	100%
2.	Fakhri	4 foto	1 foto	4 foto	100%
3.	Arie	4 foto	1 foto	4 foto	100%
4.	Rafiq	4 foto	1 foto	4 foto	100%
5.	Yayak	4 foto	1 foto	3 foto	75%
6.	Julietthania	4 foto	1 foto	3 foto	75%
7.	Nata Ferdy	4 foto	1 foto	4 foto	100%
8.	Raihan	4 foto	1 foto	3 foto	75%
9.	Fares Raja	4 foto	1 foto	4 foto	100%
10.	Dama	4 foto	1 foto	4 foto	100%



Lanjutan Tabel 4. 4 Hasil Pengujian *Face Recognition* Menggunakan *LDA* dan *Euclidean Distance*

No	Nama Partisipan	Jumlah Data Uji	Jumlah Data Latih	Data Dikenali Benar	Akurasi
12.	Farhan	4 foto	1 foto	4 foto	100%
13.	Auryn	4 foto	1 foto	4 foto	100%
14.	Fatimah	4 foto	1 foto	2 foto	50%
15.	Wulan	4 foto	1 foto	3 foto	75%
16.	Astri	4 foto	1 foto	4 foto	100%
17.	Dedi	4 foto	1 foto	2 foto	50%
18.	Putra	4 foto	1 foto	3 foto	75%
19.	Rafi	4 foto	1 foto	4 foto	100%
20.	Wira	4 foto	1 foto	2 foto	50%
21.	Arya	4 foto	1 foto	3 foto	75%
22.	Thoriq	4 foto	1 foto	4 foto	100%
23.	Akbar	4 foto	1 foto	3 foto	75%
24.	Nova	4 foto	1 foto	2 foto	50%
25.	Dian	4 foto	1 foto	3 foto	75%
26.	Salma	4 foto	1 foto	4 foto	100%
27.	Miftah	4 foto	1 foto	2 foto	50%
28.	Salsa	4 foto	1 foto	4 foto	100%
29.	Alya	4 foto	1 foto	3 foto	75%
30.	Niken	4 foto	1 foto	3 foto	75%

Total data uji sebanyak 120 citra wajah, dengan 100 data dikenali dengan benar.

Maka, akurasi total sistem adalah :

$$\text{Akurasi} = \frac{100}{120} \times 100\% = 83.33\%$$

Hasil ini menunjukkan bahwa sistem berhasil mengenali wajah partisipan dengan cukup baik dalam skenario *real-time* dengan variasi pose dan ekspresi.



4.4 Pembahasan

Berdasarkan hasil pengujian yang telah dilakukan, sistem *E-Visitor* menunjukkan kinerja yang cukup baik dalam menjalankan fungsi-fungsi utamanya. Seluruh fitur, baik dari sisi admin maupun tamu, telah berjalan sesuai dengan perancangan, termasuk proses *login*, pengisian data kunjungan, serta penyimpanan data ke dalam sistem. Hal ini menunjukkan bahwa implementasi sistem secara umum sudah dapat memenuhi kebutuhan dasar untuk proses absensi tamu berbasis *website*.

Dari sisi pengenalan wajah, sistem diuji terhadap tiga puluh partisipan dengan total seratus dua puluh citra uji. Hasil pengujian menunjukkan bahwa sistem berhasil mengenali tujuh belas dari dua puluh citra dengan benar, menghasilkan akurasi sebesar 83.33%. Nilai ini menunjukkan bahwa metode *Linear Discriminant Analysis (LDA)* yang digunakan dalam proses ekstraksi fitur wajah, serta klasifikasi dengan *Euclidean Distance* cukup efektif dalam mengenali wajah dengan ekspresi dan pose yang bervariasi.

Beberapa faktor yang dapat memengaruhi hasil pengenalan wajah antara lain pencahayaan saat proses *Login* yang berbeda dengan saat registrasi, posisi wajah yang tidak menghadap lurus ke kamera, serta ekspresi wajah yang sangat berbeda. Selain itu, kualitas kamera dari perangkat yang digunakan juga berpengaruh terhadap kualitas citra wajah yang diproses oleh sistem.

Dari hasil pengujian ini dapat disimpulkan bahwa sistem telah berhasil mengimplementasikan metode *LDA* dan *Euclidean Distance* secara fungsional dalam proses pengenalan wajah. Untuk meningkatkan akurasi pengenalan, dibutuhkan jumlah data latih yang lebih besar, serta kondisi pengambilan citra yang lebih terstandarisasi. Pengujian lanjutan terhadap lebih banyak data dan pengguna juga perlu dilakukan agar hasil yang diperoleh dapat lebih representatif dan stabil.

Secara keseluruhan, sistem ini sudah menunjukkan kinerja yang sesuai dengan tujuan awal, yaitu menciptakan sistem absensi tamu yang cepat, mandiri, dan berbasis verifikasi wajah. Penggunaan *QR Code* juga terbukti memudahkan akses ke sistem tanpa instalasi tambahan.