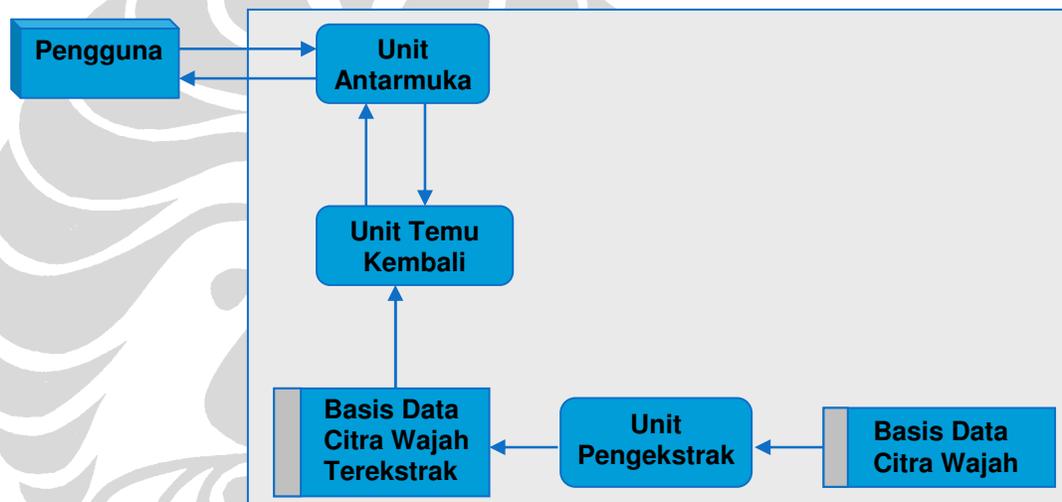


BAB 4 RANCANGAN SISTEM

Bab ini menjelaskan rancangan dan implementasi sistem. Rancangan sistem terdiri atas arsitektur sistem, rancangan modul, rancangan basis data, dan rancangan antarmuka sistem.

4.1. Arsitektur Sistem

Arsitektur sistem pengenalan wajah yang dikembangkan dapat dilihat pada gambar 4.1 di bawah ini.



Gambar 4.1 : Arsitektur Sistem

Unit Antarmuka adalah subsistem yang berfungsi untuk menerima citra wajah masukan yang diberikan oleh pengguna. Subsistem ini juga berfungsi untuk menampilkan satu citra wajah subyek yang paling mirip dengan citra wajah masukan.

Unit Pengekstrak merupakan subsistem yang berfungsi untuk melakukan ekstraksi ciri dari citra-citra wajah dalam Basis Data Citra Wajah dan menyimpan hasil ekstraksi dalam Basis Data Citra Wajah Terekstrak.

Unit Temu Kembali merupakan subsistem yang berfungsi untuk mencari subyek yang citra wajahnya paling mirip dengan citra wajah yang diberikan oleh pengguna.

Subsistem ini menggunakan hasil ekstraksi ciri citra wajah untuk menentukan citra-citra wajah yang memiliki nilai kemiripan yang tinggi terhadap citra wajah masukan. Subsistem ini akan mengirimkan citra wajah hasil pencarian ke Unit Antarmuka.

4.2. Rancangan Modul

Pada sistem pengenalan wajah ini terdapat tiga modul utama yang harus ada, yaitu modul ekstraksi ciri wajah, modul pencocokan citra wajah, dan modul Voting. Rancangan modul-modul tersebut akan dijelaskan pada subbab sebagai berikut.

4.2.1. Modul Ekstraksi Ciri Wajah

Modul ini bertanggungjawab untuk melakukan ekstraksi ciri terhadap semua citra wajah dalam basis data dan citra wajah masukan. Hasil dari ekstraksi ciri akan disimpan dan akan digunakan dalam proses pencarian citra wajah yang mirip dengan citra wajah masukan. Dalam modul ini digunakan metode PCA untuk melakukan ekstraksi ciri dengan langkah-langkah sebagai berikut :

1. Menyatakan citra wajah x_i sebagai vektor baris dengan dimensi n dan berisi nilai dari piksel dalam citra wajah x_i

$$x_i = [x_{i1} \ x_{i2} \ x_{i3} \ \dots \ x_{in}] \quad (4.1)$$

dimana $i = 1, 2, \dots, N$, $n = \text{lebar} \times \text{tinggi}$ dari citra wajah, N adalah jumlah citra wajah dalam basis data.

2. Menghitung rata-rata dari vektor citra wajah dalam basis data

$$\mu = \frac{1}{N} \sum_{k=1}^N x_k \quad (4.2)$$

3. Mencari selisih antara rata-rata basis data citra wajah, μ , dengan citra wajah x_i

$$\Phi_i = x_i - \mu \quad (4.3)$$

dan disimpan ke dalam matriks A.

$$A = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \dots \\ \Phi_N \end{pmatrix} \quad (4.4)$$

4. Mencari matriks kovarian atau *total scatter* dari matriks masukan A

$$S_T = A^T A \quad (4.5)$$

5. Mencari vektor *eigen* dan nilai *eigen* dari matriks kovarian, dan diurutkan berdasarkan nilai *eigen* terbesar

$$[EV, EL] = \text{eigen}(S_T) \quad (4.6)$$

$$\text{dimana } EL = [d_1; d_2; d_3; \Lambda \Lambda \Lambda ; d_n] \quad (n \times 1)$$

$$\text{dengan } d_1 > d_2 > d_3 > \Lambda \Lambda \Lambda > d_n$$

$$\text{dan } EV = \begin{bmatrix} w_{11} & w_{12} & \Lambda & \Lambda & \Lambda & w_{1n} \\ M & M & & & & M \\ w_{n1} & w_{n2} & \Lambda & \Lambda & \Lambda & w_{nm} \end{bmatrix} \quad (n \times n)$$

6. Menentukan *m* vektor *eigen* yang akan digunakan berdasarkan *m* nilai *eigen* terbesar, dimana $m < n$

$$m = \min_r \left\{ \frac{\sum_{i=1}^r d_i}{\sum_{i=1}^N d_i} > \Theta \right\}, 0 < \Theta \leq 1 \quad (4.7)$$

7. Mereduksi vektor *eigen* sesuai dengan dimensi *m*

$$EV' = \begin{bmatrix} w_{11} & w_{12} & \Lambda & \Lambda & \Lambda & w_{1m} \\ M & M & & & & M \\ w_{n1} & w_{n2} & \Lambda & \Lambda & \Lambda & w_{nm} \end{bmatrix} \quad (n \times m) \quad (4.8)$$

dan matriks transformasi atau Eigenface *W* adalah

$$W = EV'^T \quad (m \times n) \quad (4.9)$$

8. Mentransformasi vektor citra wajah ke vektor ciri

$$ER = AW^T \quad (N \times m) \quad (4.10)$$

Prosedur yang berperan dalam proses ekstraksi ciri adalah :

a. Prosedur Rata-rata

Prosedur ini berfungsi untuk menghitung nilai rata-rata dari semua citra wajah dalam basis data, sebagaimana dirumuskan pada persamaan (4.2).

b. Prosedur Selisih

Prosedur ini berfungsi untuk menghitung selisih antara tiap nilai piksel dalam citra wajah dengan nilai rata-rata seluruh citra wajah. Prosedur ini dirumuskan dengan persamaan (4.3) dan (4.4).

c. Prosedur Matriks Kovarian

Prosedur ini berfungsi untuk memperoleh matriks kovarian dari matriks yang merepresentasikan seluruh citra dalam basis data. Perumusan prosedur ini dituliskan pada persamaan (4.5).

d. Prosedur *Eigen*

Prosedur ini berfungsi untuk mencari vektor *eigen* dan nilai *eigen* dari matriks kovarian seluruh citra wajah dalam basis data. Prosedur ini mengimplementasikan rumus pada persamaan (4.6).

e. Prosedur Transformasi

Prosedur ini berfungsi untuk mentransformasikan vektor *eigen* dari seluruh citra wajah dalam basis data dengan dimensi tinggi ke ruang ciri dengan dimensi yang lebih rendah. Dalam prosedur ini, dilibatkan proses penentuan dimensi baru hasil reduksi dan proses perolehan vektor ciri berdasarkan dimensi baru hasil reduksi. Prosedur ini berusaha mengimplementasikan rumus pada persamaan (4.7) sampai dengan (4.10).

4.2.2. Modul Pencocokan Wajah

Modul ini bertanggungjawab untuk melakukan pencarian citra wajah dalam basis data dengan nilai kemiripan yang cukup tinggi dengan citra wajah masukan. Beberapa prosedur yang berperan dalam proses pencocokan ini adalah :

a. Prosedur Euclidean

Prosedur ini berfungsi untuk menghitung jarak antara citra wajah masukan dengan setiap citra wajah dalam basis data. Hasil dari prosedur ini adalah suatu matriks yang merepresentasikan jarak antara dua buah citra secara terurut. Prosedur ini berusaha mengimplementasikan algoritma Jarak Euclidean yang dijelaskan pada bab 2.

b. Prosedur Normalisasi

Prosedur ini berfungsi untuk menormalisasi jarak antara dua buah citra wajah sehingga berada dalam rentang $[0,1]$. Hasil normalisasi ini merepresentasikan tingkat kemiripan antara dua buah citra wajah.

c. **Prosedur Pencocokan**

Prosedur ini bertujuan untuk memperoleh top- N citra wajah dengan nilai kemiripan yang cukup tinggi terhadap citra wajah masukan.

4.2.3. Modul Voting

Modul ini bertanggungjawab untuk melakukan proses Voting terhadap n-top citra wajah sehingga menghasilkan satu citra wajah sebagai hasil keluaran. Beberapa prosedur yang berperan dalam proses Voting ini adalah :

a. **Prosedur Penghitungan Bobot**

Prosedur ini berfungsi untuk memberikan bobot dan menghitung bobot dari setiap subyek yang ada di himpunan citra wajah hasil.

b. **Prosedur Pemilihan Citra Wajah Hasil**

Prosedur ini berfungsi untuk mengambil keputusan dalam memilih citra wajah hasil. Pengambilan keputusan tersebut adalah berdasarkan bobot setiap subyek yang telah diberikan pada Prosedur Penghitungan Bobot.

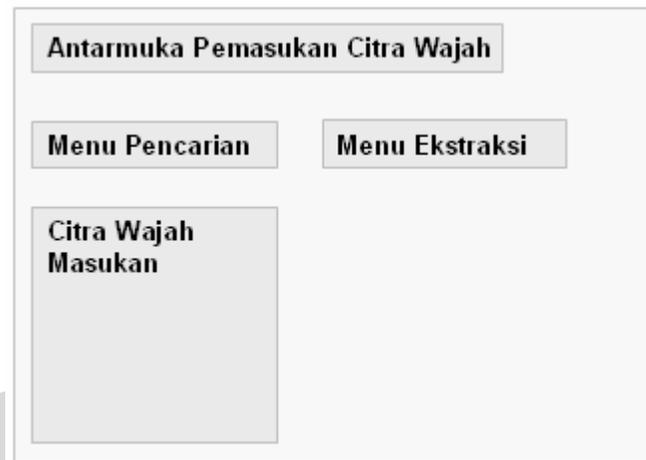
4.3. Rancangan Basis Data

Dalam sistem ini, digunakan suatu sistem file untuk menyimpan basis data citra wajah. Bentuk dari sistem file ini adalah suatu folder yang terdiri atas seluruh citra wajah yang digunakan sebagai basis data. Hasil ekstraksi ciri dari seluruh citra wajah disimpan dalam suatu file dengan format teks (.txt).

4.4. Rancangan Antarmuka

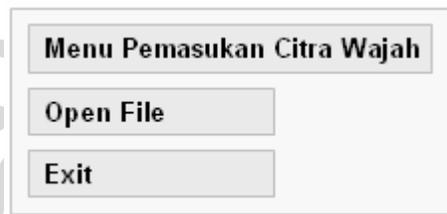
Sistem yang akan dikembangkan adalah suatu sistem yang berbasis GUI. Antarmuka dari sistem yang akan dikembangkan adalah sebagai berikut :

1. Antarmuka Utama Sistem



Gambar 4.2 : Antarmuka Utama dari Sistem

2. Antarmuka Pemasukan Citra Wajah



Gambar 4.3 : Antarmuka Pemasukan Citra

3. Antarmuka Hasil Pencarian



Gambar 4.4 : Antarmuka Hasil Pencarian

BAB 5

IMPLEMENTASI SISTEM

Bab ini akan menjelaskan mengenai implementasi dari sistem yang akan dikembangkan, yaitu meliputi spesifikasi sistem, implementasi prosedur, dan implementasi antarmuka sistem.

5.1. Spesifikasi Sistem

Pada subbab ini akan dijelaskan tentang komponen-komponen yang membangun arsitektur dari sistem, meliputi komponen perangkat keras dan perangkat lunak.

5.1.1. Perangkat Keras

Sistem Pengenalan Wajah ini dikembangkan pada komputer dengan spesifikasi perangkat keras sebagai berikut :

Processor : Intel(R) Pentium(R) 4 CPU 1.6 GHz (2CPUs)

Memory : 512MB RAM

5.1.2. Perangkat Lunak

Perangkat lunak yang digunakan untuk mengembangkan sistem ini adalah sebagai berikut :

- Sistem Operasi : Microsoft Windows XP Professional Version 2002
- Lingkungan Pengembangan (IDE) : Microsoft Visual C++ 6.0
- Library : OpenCV [25]. *Library* ini menyediakan fungsi-fungsi untuk memproses citra, melakukan perhitungan matriks, dan lain-lain.

5.2. Implementasi Prosedur

Berikut ini adalah beberapa prosedur yang digunakan dalam implementasi sistem pengenalan wajah.

5.2.1. Prosedur Rata-rata

```
Mean(meanMatrix, inputMatrix, sigma, miu)
  size := column(inputMatrix);
  N := rows(inputMatrix);
  r := rows(meanMatrix);
  for i = 0 : size
    for j = 0 : N
      sigma ← sigma + inputMatrix(j,i)
  miu = sigma/N;
  meanMatrix(r,i) ← miu
```

5.2.2. Prosedur Selisih Rata-rata

```
Diff(inputMatrix, meanMatrix, newInputMatrix)
  size := column(inputMatrix);
  N := rows(inputMatrix);
  r := rows(meanMatrix);
  for i = 0 : N
    for j = 0 : size
      diff ← inputMatrix(i,j)-meanMatrix(r,j)
      newMatrix(i,j) ←diff
```

5.2.3. Prosedur Matriks Kovarian

```
CovMat(newMatrix, covMatrix)
  transMatrix ← transpose(newMatrix)
  covMatrix ← newMatrix * transMatrix
```

5.2.4. Prosedur Eigen

```
Eigen(covMatrix, eigenMatrix, egvalMatrix)
[eigenMatrix, egvalMatrix] ← eigen(covMatrix)
```

5.2.5. Prosedur Transformasi

```
Transform(eigenSpaceMatrix, newMatrix, eigenMatrix)
m;
redEigenMatrix ← reduction(eigenMatrix, m)
eigenSpaceMatrix ← newMatrix * redEigenMatrix
```

5.2.6. Prosedur Euclidean

```
Euclidean (eigenSpaceMatrix, inputMatrix, distanceMatrix)
size := column(eigenSpaceMatrix);
N := rows(eigenSpaceMatrix);
r := rows(inputMatrix);
for i = 0 : N
  for j = 0 : size
    temp ← eigenSpaceMatrix(i, j) - inputMatrix(r, j)
    temps ← temps + pow(temp, 2)
  root ← sqrt(temps)
  distanceMatrix(i) ← root
```

5.2.7. Prosedur Normalisasi Jarak

```
Normalize(normalMatrix, distanceMatrix)
size := column(distanceMatrix);
N := rows(distanceMatrix);
for i = 0 : N
  for j = 0 : size
    sum ← sum + distanceMatrix(i, j)
for i = 0 : N
  for j = 0 : size
    normal ← (sum - distanceMatrix(i, j)) / sum
```

```
normalMatrix(i,j) ← normal
```

5.2.8. Prosedur Pencocokan

```
Matching (eigenSpaceMatrix, inputMatrix)  
    Euclidean(eigenSpaceMatrix, inputMatrix,  
            distanceMatrix)  
    Normalize(normalMatrix, distanceMatrix)  
    for i = 0 : 10  
        showImage(i)
```

5.2.9. Prosedur Penghitungan Bobot

```
onDisplayImage()  
for i=0 : i<10  
    for j=0 : j<row  
        char buf[3]  
        int index=voted(kode);  
        if(index== -1)  
            vote[count][0]=1;  
            vote[count][1]=kode;  
            vote[count][2]=j;  
            ++count;  
        else vote[index][0]=vote[index][0]+1;
```

5.2.10. Prosedur Pemilihan Citra Wajah Hasil

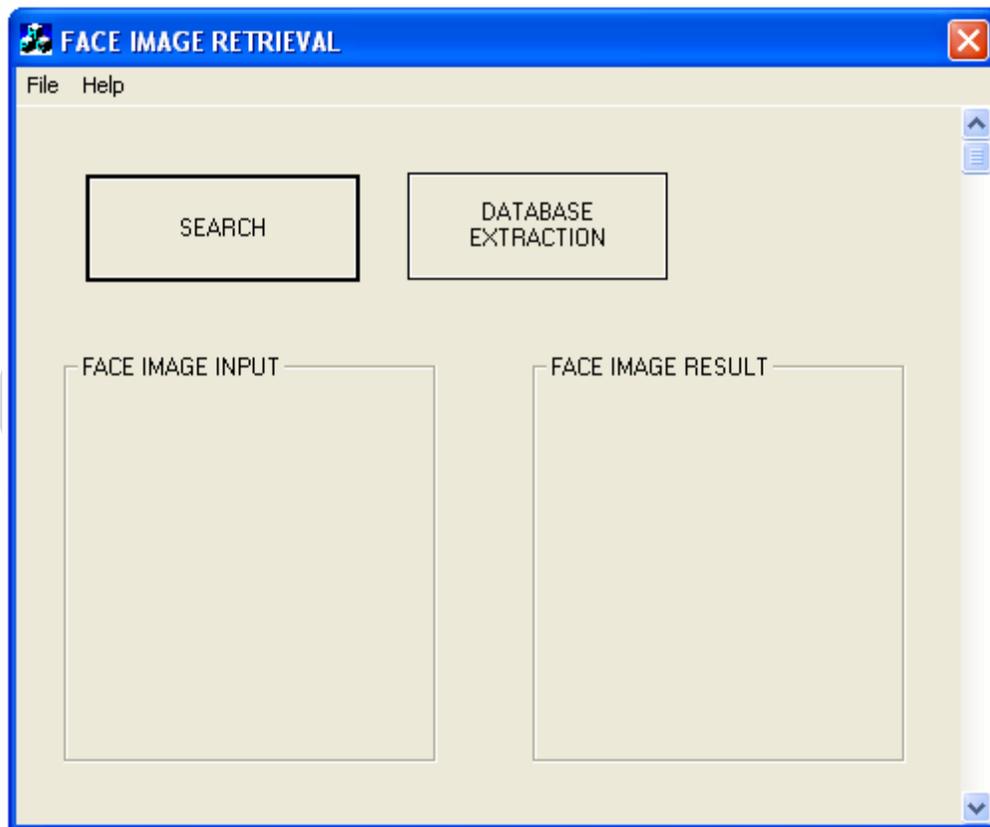
```
getMax()  
    indexHasil :=0  
    for i=1 : i<10  
        if(vote[i][0]>vote[indexHasil][0])  
            indexHasil=i;  
    return vote[indexHasil][2];
```

5.3. Antarmuka

Berikut ini adalah tampilan antarmuka dari Sistem Pengenalan Wajah yang berbasis GUI.

5.3.1. Antarmuka Utama Sistem

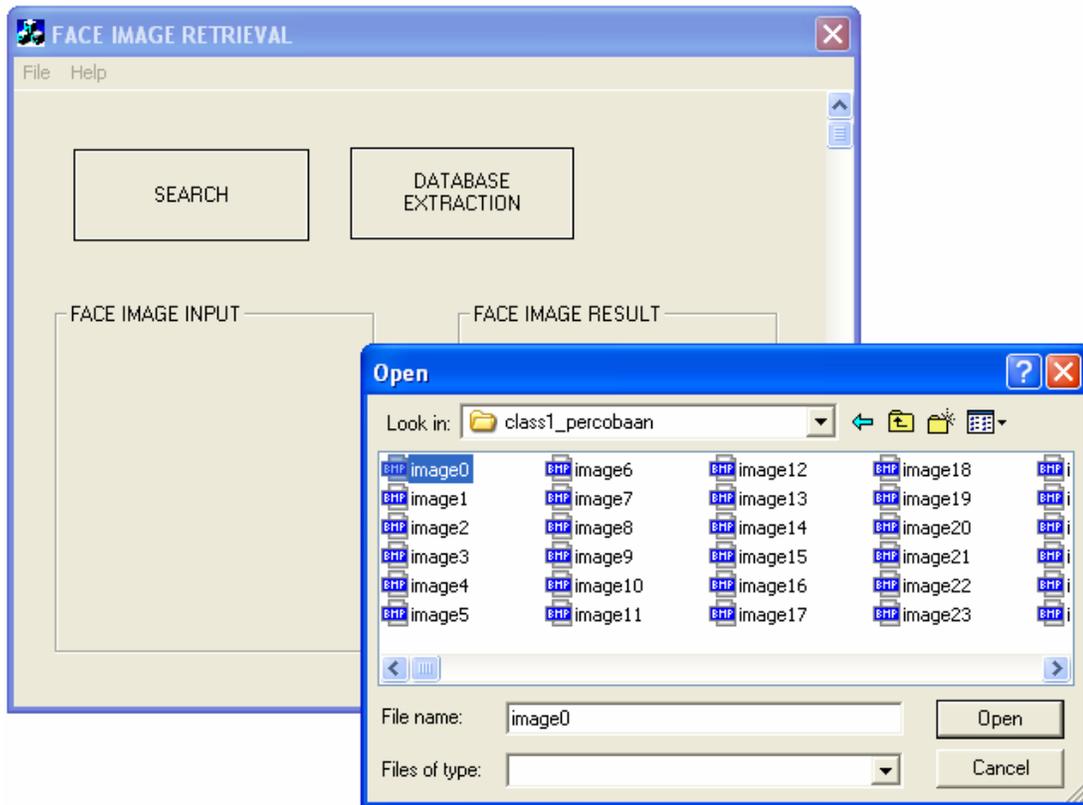
Tampilan antarmuka utama sistem dapat dilihat pada Gambar 5.1 di bawah ini.



Gambar 5.1 : Antarmuka Utama Sistem

5.3.2. Antarmuka Pemasukan Citra Wajah

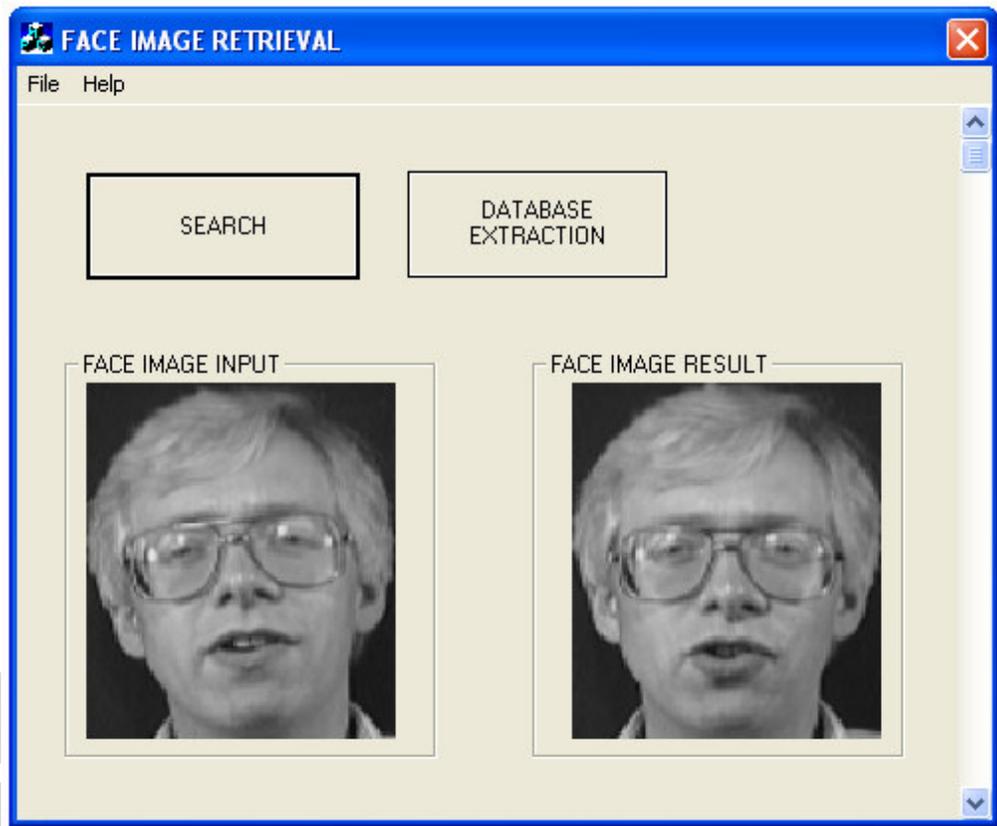
Tampilan antarmuka pemasukkan citra wajah dapat dilihat pada Gambar 5.2.



Gambar 5.2 : Antarmuka Pemasukan Citra Wajah

5.3.3. Antarmuka Hasil Pencarian

Tampilan antarmuka hasil pencarian citra wajah dapat dilihat pada Gambar 5.3.



Gambar 5.3 : Antarmuka Hasil Pencarian

BAB 6

UJI COBA DAN ANALISIS

Bab ini menjelaskan mengenai data uji coba yang digunakan, skenario pengujian sistem, hasil pengujian sistem, dan analisa hasil pengujian sistem.

6.1. Data Uji Coba

Pada penelitian ini digunakan dua basis data citra wajah, yaitu:

1. Citra wajah dari *AT&T Laboratory, Cambridge University* [10]

Pada citra wajah dari *AT&T Laboratory*, terdapat 300 citra wajah, yang terdiri atas 30 kelas wajah. Masing-masing kelas terdiri atas 10 citra wajah dengan pose yang berbeda untuk orang yang sama. Citra wajah *AT&T Laboratory* dapat dilihat pada Lampiran 1.

2. Citra wajah dari *UMIST*[11].

Pada citra wajah *UMIST*, terdapat 200 citra wajah, yang terdiri atas 20 kelas wajah. Masing-masing kelas terdiri atas 10 citra wajah dengan pose yang berbeda untuk orang yang sama. Citra wajah *UMIST* memiliki sudut pose yang lebih besar dibandingkan citra wajah *AT&T Laboratory*. Citra wajah *UMIST* dapat dilihat pada Lampiran 2.

Penggunaan dua jenis citra wajah tersebut, bertujuan untuk membandingkan akurasi setiap skenario Voting yang dikembangkan, karena skenario Voting yang sama dapat memberikan tingkat akurasi yang berbeda saat diterapkan pada basis data citra wajah yang berbeda.

6.2. Skenario Uji Coba

Dalam melakukan pengujian, terdapat dua istilah yang akan digunakan yaitu citra wajah pelatihan dan citra wajah pengujian. Data training adalah himpunan citra wajah yang akan diekstrak dan menjadi basis data citra wajah. Sedangkan data pengujian adalah citra wajah masukan.

Untuk menguji kinerja dari sistem temu kembali citra pada wajah, dilakukan tiga jenis pengujian, yaitu pada citra wajah *AT&T Laboratory*, *UMIST*, dan gabungan keduanya, dimana pada setiap citra wajah tersebut diterapkan skenario pengujian yang sama. Skenario pengujian sistem yang dilakukan adalah sebagai berikut.

1. Pada pengujian pertama, digunakan perbandingan citra pelatihan dan pengujian sebesar 7:3.
2. Pada pengujian kedua, digunakan perbandingan citra pelatihan dan pengujian sebesar 5:5.
3. Pada pengujian ketiga, digunakan perbandingan citra pelatihan dan pengujian sebesar 3:7.

Tujuan dari ketiga skenario pengujian tersebut adalah untuk mengetahui kemampuan sistem dalam mengenali wajah orang yang sama, akan tetapi dalam citra wajah yang berbeda.

6.3. Hasil Pengujian Sistem

Pada subbab ini akan dijelaskan hasil pengujian berdasarkan skenario pengujian yang dijelaskan pada subbab sebelumnya.

6.3.1. Skenario 1

Akurasi hasil pencarian citra wajah menggunakan skenario 1 dapat dilihat pada Tabel 6.1.

Tabel 6.1: Data Akurasi Skenario 1

Basis Data			Fungsi Kemiripan	Pengurangan Tingkat Kesalahan Terhadap Metode Jarak Euclidean
AT&T	UMIST	Gabungan		
98.9%	86.7%	94%	Jarak Euclidean	0
90%	63.3%	79.3%	Skenario Voting 1	0
93.3%	70%	84%	Skenario Voting 2	0
95.6%	73.3%	88%	Skenario Voting 3	0
83.3%	56.7%	72.7%	Skenario Voting 4	0
87.8%	66.7%	79.3%	Skenario Voting 5	0
94.4%	71.7%	85.3%	Skenario Voting 6	0

Berdasarkan hasil pencarian tersebut dapat diambil kesimpulan bahwa metode Jarak Euclidean memberikan akurasi yang paling tinggi dibandingkan semua fungsi kemiripan yang menggunakan metode Voting. Sedangkan skenario Voting yang memberikan akurasi tertinggi adalah skenario Voting 3, dan skenario Voting 3 ini menurunkan akurasi metode Jarak Euclidean sebesar 6%.

Pada percobaan menggunakan skenario 1 ini, semua skenario Voting yang dibuat tidak berhasil mengurangi kesalahan yang terjadi pada penggunaan metode Jarak Euclidean.

6.3.2. Skenario 2

Hasil pencarian untuk suatu percobaan dalam skenario 2 ini dapat dilihat pada Tabel 6.2.

Tabel 6.2 : Data Akurasi Skenario 2

Basis Data			Fungsi Kemiripan	Pengurangan Tingkat Kesalahan Terhadap Metode Jarak Euclidean
AT&T	UMIST	Gabungan		
92%	68.8%	80.4%	Jarak Euclidean	0
74.45	59.2%	66.8%	Skenario Voting 1	4.1%
80.8%	69%	70.4%	Skenario Voting 2	2.1%
84.8%	68%	74.8%	Skenario Voting 3	4.1%
65.6%	57.6%	61.6%	Skenario Voting 4	6.1%
72%	59.2%	65.6%	Skenario Voting 5	2.1%
83.2%	62.4%	72.8%	Skenario Voting 6	6.1%

Pada percobaan menggunakan skenario 2, pola akurasi yang dihasilkan tidak jauh berbeda dengan skenario 1, dimana metode Jarak Euclidean memberikan akurasi yang paling tinggi, dan skenario Voting 3 memberikan akurasi tertinggi diantara semua skenario Voting yang dibuat. Skenario Voting 3 menurunkan akurasi metode Jarak Euclidean sebesar 5.6%.

Pada percobaan menggunakan skenario 2 ini, skenario Voting 4 dan 6 mengurangi kesalahan yang terjadi pada penggunaan metode Jarak Euclidean hingga 6.1%. sedangkan skenario Voting 3 yang merupakan skenario Voting dengan akurasi tertinggi, mengurangi kesalahan pada metode Jarak Euclidean sebesar 4.1%

6.3.3. Skenario 3

Hasil pencarian untuk suatu percobaan dalam skenario 3 ini dapat dilihat pada Tabel 6.3.

Tabel 6.3: Data Akurasi Skenario 3

Basis Data			Fungsi Kemiripan	Pengurangan Tingkat Kesalahan Terhadap Metode Jarak Euclidean
AT&T	UMIST	Gabungan		
86.1%	60%	75.7%	Jarak Euclidean	0
73.8%	53.6%	65.7%	Skenario Voting 1	9.4%
77.6%	55%	68.6%	Skenario Voting 2	9.4%
82.9%	57.9%	72.9%	Skenario Voting 3	5.9%
67.1%	55.7%	62.6%	Skenario Voting 4	9.4%
69%	53.6%	62.9%	Skenario Voting 5	8.2%
77.1%	54.3%	68%	Skenario Voting 6	5.9%

Pada percobaan menggunakan skenario 3, pola akurasi yang dihasilkan tidak jauh berbeda dengan skenario 1 dan skenario 2, dimana metode Jarak Euclidean memberikan akurasi yang paling tinggi, dan skenario Voting 3 memberikan akurasi tertinggi diantara semua skenario Voting yang dibuat. Skenario Voting 3 menurunkan akurasi metode Jarak Euclidean sebesar 2.8%.

Pada percobaan menggunakan skenario 3 ini, skenario Voting 4 dan 1 berhasil mengurangi kesalahan yang terjadi pada penggunaan metode Jarak Euclidean sebesar 9.4%. sedangkan skenario Voting 3 yang merupakan skenario Voting dengan akurasi tertinggi, berhasil mengurangi kesalahan pada metode Jarak Euclidean sebesar 5.9%.

6.4. Analisa

Berdasarkan hasil yang diperoleh dari ketiga skenario pengujian, perbedaan sudut posisi citra wajah mempengaruhi hasil pencarian citra wajah. Semakin besar sudut antara citra wajah masukan dan citra wajah yang ada di basis data, maka akurasi pencariannya akan semakin rendah. Hal ini bisa dilihat pada 3 skenario percobaan, dimana akurasi pada basis data UMIST lebih rendah dibandingkan akurasi pada basis data AT&T. Hal tersebut disebabkan sudut pose citra wajah pada basis data UMIST lebih besar daripada citra wajah pada basis data AT&T.

Dari ketiga skenario percobaan yang dilakukan, skenario Voting 3 adalah skenario Voting yang berhasil memberikan akurasi tertinggi dibandingkan skenario Voting lainnya. Meskipun demikian, berdasarkan ketiga skenario percobaan yang dilakukan, skenario Voting 3 menurunkan rata-rata akurasi metode Jarak Euclidean hingga 4.8%.

Penurunan akurasi yang terjadi pada semua skenario Voting tersebut, disebabkan citra wajah subyek masukan yang diharapkan akan muncul pada n-top citra wajah ternyata tidak cukup untuk membuat akurasi skenario Voting melampaui akurasi metode Jarak Euclidean. Kesalahan pada skenario Voting disebabkan citra wajah subyek masukan berada pada urutan yang terlalu jauh dari n-top citra wajah, sehingga citra wajah tersebut tidak bisa berkontribusi untuk memberikan hasil pencarian yang benar.