

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/315330639>

RANCANG BANGUN SISTEM PENGENALAN WAJAH DENGAN METODE PRINCIPAL COMPONENT ANALYSIS

Article · August 2016

DOI: 10.36341/rabit.v1i2.25

CITATIONS

4

READS

2,716

2 authors, including:



[Salamun M. Kom](#)

Universitas Abdurrah

3 PUBLICATIONS 6 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Aplikasi Mobile View project

RANCANG BANGUN SISTEM PENGENALAN WAJAH DENGAN METODE PRINCIPAL COMPONENT ANALYSIS

Salamun¹, Firman Wazir²

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Abdurrah Pekanbaru
Jl. Riau Ujung No. 73 Pekanbaru
Telp. 0761-38762
salamun@univrab.ac.id, Firmanwazir@gmail.com

ABSTRAK

Wajah merupakan salah satu ukuran fisiologis yang paling mudah dan sering digunakan untuk membedakan identitas individu yang satu dengan yang lainnya. Proses pengenalan wajah ini menggunakan informasi mentah dari *pixel* citra yang dihasilkan melalui *camera* yang kemudian direpresentasikan dalam metode *Principal Components Analysis*. Adapun cara kerja metode *Principal Components Analysis* adalah dengan menghitung dari rata-rata *flatvector pixel* dari gambar-gambar yang sudah disimpan dalam suatu *database*, dari rata-rata *flatvector* tersebut akan didapatkan nilai *eigenface* masing-masing gambar dan kemudian akan dicari nilai *eigenface* terdekat dari gambar citra wajah yang ingin dikenali. Hasil pengujian menunjukkan tingkat keberhasilan pengenalan wajah secara keseluruhan sebesar 82,27% dengan data wajah sebanyak 130 gambar.

Kata Kunci : PCA, Eigenface, Flatvector, Pengenalan Wajah.

ABSTRACT

The face was one of the easiest physiological measure and often used to distinguish the identity of one individual to another. The face recognition process using the raw information from the pixel image produced by the camera was then represented in the Principal Components Analysis method. As for how the Principal Components Analysis method was to calculate the average flatvector pixel of the images that have been stored in a database, from an average of flatvector will be obtained tilapia Eigenface each image and then look for the value Eigenface closest pictures face image that you want to identify. The test results show a success rate of face recognition as a whole by 82,27 % with the data face as many as 130 images.

Keywords : PCA, Eigenface, Flatvector, Face Recognition.

I. PENDAHULUAN

1. Latar Belakang Masalah

Teknologi di bidang informasi khususnya dengan menggunakan komputer yang telah tumbuh semakin pesat, berbagai macam aplikasi telah dirancang untuk mencari informasi tentang data diri seseorang. Salah satu aplikasi yang cukup dikenal dalam hal ini adalah aplikasi pengenalan wajah (*face recognition*). Aplikasi *face recognition* pada saat ini banyak dikembangkan karena dapat diaplikasikan di berbagai bidang permasalahan, salah satu contohnya adalah pengenalan kriminal, *system security*. Sistem identifikasi seseorang berdasarkan wajah atau biasa disebut sistem pengenalan wajah,

memang baru-baru ini mulai digunakan pada bidang biometrika, apakah sistem ini bisa menjadi solusi alternatif untuk diterapkan pada sistem keamanan.

Sama halnya dengan sidik jari dan retina mata manusia, wajah bersifat nonidentik karena setiap orang memiliki bentuk dan kontur wajah yang berbeda-beda, sehingga wajah bisa dijadikan alat atau model untuk dipakai dalam mengidentifikasi seseorang. Sistem pengenalan wajah adalah suatu sistem yang membuat sebuah mesin dapat mengenali wajah seseorang sesuai dengan gambar wajah yang telah dilatih dan disimpan di dalam mesin tersebut.

Untuk membangun sistem pengenalan wajah yang baik agar dapat diterapkan pada sistem identifikasi seseorang, sistem tersebut harus memenuhi beberapa kriteria diantaranya keakuratan pengenalan dan kecepatan pengenalan. Dengan adanya masalah tersebut maka diperlukan suatu analisis terhadap system pengenalan wajah, oleh karena itu dilakukan penelitian tentang analisis sistem pengenalan wajah dengan metode *principal component Analysis*. Metode *Principal Component Analysis* (PCA) merupakan salah satu metode implementasi yang banyak digunakan untuk pengenalan pola dan *image coding*.

2. Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka penulis dapat merumuskan permasalahan yang diteliti, yaitu :

1. Bagaimana dapat mengenali suatu pola, khususnya pola wajah ?
2. Bagaimana membangun aplikasi pengenalan wajah menggunakan metode *Principal Component Analysis* ?
3. Apakah pengenalan wajah dengan menggunakan metode *Principal Component Analysis* dapat mencapai presentase keberhasilan diatas 50% ?
4. Bagaimana melakukan proses pengenalan citra wajah menggunakan metode *Principal Component Analysis*?

3. Batasan Masalah

Untuk menjaga fokus dari penelitian ini, maka beberapa batasan yang diberikan adalah sebagai berikut :

1. Wajah yang akan dideteksi adalah wajah yang menghadap ke depan (frontal), dalam posisi tegak, dan tidak terhalangi sebagian oleh objek lain.
2. Jarak antara wajah dengan kamera kurang lebih 30 cm sampai 50 cm.
3. Pencahayaan di lingkungan sekitar wajah harus terang (minimal ada sumber cahaya yang menghadap ke wajah yang akan ditangkap oleh kamera).

4. Gambar di ambil secara *realtime*
5. Bahasa yang akan digunakan adalah bahasa *C#* dan *Library EmguCV*
6. *Capture device* yang digunakan adalah *Android-camera*.

4. Tujuan

Penelitian yang dilakukan memiliki tujuan sebagai berikut :

1. Mengukur tingkat keberhasilan pengenalan wajah dengan menggunakan algoritma *Principal Component Analysis* (PCA) sesuai dengan batasan yang telah ditentukan.
1. Menghasilkan aplikasi pengenalan wajah menggunakan metode *Principal Component Analysis*.

5. Manfaat

Beberapa manfaat yang diperoleh dari penelitian yang dilakukan :

1. Dapat mencari informasi orang yang melakukan kejahatan atau pelanggaran melalui foto yang telah di scan.
2. Dapat mengukur kecepatan dalam pengenalan wajah menggunakan metode *principal componen tAnalysis*.
3. Diharapkan hasil penelitian ini dapat membantu pengembangan aplikasi-aplikasi yang lebih kompleks seperti sistem keamanan menggunakan wajah.

II. STUDI PUSTAKA

1. Pengenalan Wajah

Pengenalan wajah adalah suatu metoda pengenalan yang berorientasi pada wajah. Pengenalan ini dapat dibagi menjadi dua bagian yaitu : Dikenali atau tidak dikenali, setelah dilakukan perbandingan dengan pola yang sebelumnya disimpan di dalam *database* (Marti, 2010). Secara umum, sistem pengenalan citra wajah dibagi menjadi 2 jenis, yaitu sistem *feature based* dan *image-based*. Pada sistem pertama digunakan fitur yang diekstraksi dari komponen citra wajah (mata, hidung, mulut, dll) yang kemudian

hubungan antara fitur-fitur tersebut dimodelkan secara geometris. Sedangkan sistem kedua menggunakan informasi mentah dari piksel citra yang kemudian direpresentasikan dalam metode tertentu, yang kemudian digunakan untuk klasifikasi identitas citra (Fatta, 2009).

2. Pendeteksi Wajah

Proses pendeteksi wajah ini bekerja dengan cara memeriksa citra yang dimasukkan, apakah memiliki citra wajah atau tidak, jika memiliki, maka akan dilakukan pemisahan dengan cara memotong citra wajah dari latar belakang citra yang dimasukkan. Jika masukan berbentuk video, proses yang dilakukan adalah proses *face tracking*. Secara umum, proses *face tracking* dan proses pendeteksian wajah mempunyai fungsi yang sama. Perbedaannya terletak pada proses pendeteksian saja, jika pada masukan berbentuk citra, sistem berjalan *offline* sehingga dapat menggunakan proses pendeteksian wajah, sedangkan pada masukan video, sistem berjalan secara *online* atau *real-time* yang membutuhkan pendeteksian secara langsung maka proses yang digunakan adalah proses *face tracking*.

3. Penyelarasan Wajah

Pada proses pendeteksian wajah, citra wajah yang didapatkan masih berupa perkiraan kasar atau masih memiliki kualitas yang cukup buruk seperti ukuran yang berbeda dengan ukuran normal, faktor pencahayaan yang kurang atau lebih, kejelasan citra yang buruk dan sebagainya. Maka perlu diadakannya proses penyelarasan. Proses penyelarasan wajah merupakan proses yang bertujuan untuk menormalisasi wajah dari citra wajah yang didapatkan dari proses pendeteksian wajah. Proses ini terdiri dari tahapan-tahapan sebagai berikut:

1. Grayscale

Grayscale citra merupakan tahapan pertama dari proses penyelarasan, pada tahap ini terjadi pengkonversian citra

warna RGB menjadi citra berwarna abu. Citra warna RGB terdiri dari 3 parameter warna yaitu merah (*red*), hijau (*green*) dan biru (*blue*), jika citra warna RGB ini dimasukan ke dalam proses ekstraksi, maka proses tersebut akan sulit untuk dilakukan karena citra RGB terdiri dari 3 parameter, oleh karena itu diperlukan penyamaan parameter yaitu dengan melakukan tahap *grayscale* ini.

2. Pemotongan

Pada tahapan ini terjadi pemotongan citra yang memisahkan citra wajah dengan citra masukannya, tujuannya untuk mengambil citra yang hanya diperlukan untuk proses ekstraksi, dalam hal ini adalah citra wajah dan membuang citra lain yang tidak diperlukan. Dimensi citra yang dipotong disesuaikan dengan dimensi dari proses segmentasi atau pengkotakan objek wajah yang dilakukan pada proses pendeteksian wajah.

3. Resizing (tahap normalisasi dimensi citra)

Pada tahap *resizing* citra, terjadi proses normalisasi dimensi citra wajah, yaitu proses pembesaran atau pengecilan dimensi citra wajah menjadi dimensi yang telah ditentukan. Tujuannya, untuk menyamakan dimensi wajah dari tiap citra yang dimasukkan, sehingga pada proses ekstraksi citra nanti tidak ada perbedaan dimensi dari matriks data citra wajah.

4. Equalizing (tahap koreksi tingkat kecerahan citra)

Tahap ini adalah tahapan terakhir dari proses penyelarasan, yang tujuannya untuk memperjelas nilai *histogram* dari citra wajah hasil tahapan-tahapan sebelumnya.

4. Ekstraksi Wajah

Ekstraksi fitur adalah proses untuk mendapatkan ciri-ciri pembeda yang membedakan suatu sampel wajah dari sampel wajah yang lain (Damayanti et al, 2010).

Ekstraksi fitur merupakan proses yang berfungsi untuk mendapatkan informasi yang efektif dan berguna untuk membedakan wajah-wajah orang yang berbeda dari citra wajah yang telah diselaraskan. Proses ini dilakukan menggunakan algoritma-algoritma ekstraksi seperti *principal component analysis* (PCA), *linear discriminant analysis* (LDA), *independent component analysis* (ICA) dan sebagainya. Informasi yang didapatkan dari ekstraksi fitur disebut vektor fitur, yaitu bentuk dasar pencarian citra berbasis konten, yang menangkap properti citra seperti warna dan tekstur.

5. Penyimpanan Fitur Wajah

Proses penyimpanan fitur merupakan tahapan terakhir dari proses pelatihan citra wajah. Proses ini berfungsi untuk menyimpan fitur hasil ekstraksi citra wajah yang ada di dalam database ke dalam sebuah *file* berekstensi *.xml. *File* inilah yang nantinya akan digunakan untuk proses pencocokan antara citra wajah yang diuji dengan hasil ekstraksi fitur yang terdapat pada *file* ini.

6. Pencocokan Fitur Wajah

Pencocokan fitur merupakan proses perbandingan fitur yang telah diekstrak dari citra uji dengan fitur citra wajah dari *database*, yang sebelumnya telah melalui proses pelatihan citra.

Dari proses perbandingan fitur tersebut akan menghasilkan nilai jarak terdekat yang menandakan nilai fitur citra uji hampir menyamai dengan fitur citra latih. Nilai jarak ini akan menjadi nilai masukan untuk nilai kemiripan citra.

Nilai kemiripan citra merupakan nilai tingkat kemiripan citra uji dengan citra latih, semakin besar nilainya menandakan bahwa orang yang sedang diamati adalah orang yang sama dengan orang yang citra wajahnya telah disimpan dalam *database*. Nilai kemiripan ini sebelum mengeluarkan *output* hasil pengenalan wajah akan melalui proses *threshold* terlebih dahulu.

Proses *threshold* pengenalan wajah adalah proses penyaringan nilai kemiripan citra, dimana jika nilai kemiripan berada dibawah *threshold* yang telah ditentukan maka output hasil pengenalan tidak akan ditampilkan. Jika, berada diatas output hasil pengenalan akan ditampilkan. Tujuannya agar output hasil pengenalan yang ditampilkan bernilai benar, karena memiliki nilai kemiripan yang tinggi. Setelah, proses-proses tersebut dilakukan proses terakhir adalah proses pencarian tingkat keakuratan. Keakuratan adalah kondisi yang menunjukkan kebenaran atau ketepatan suatu objek yang diamati, dalam sistem pengenalan ini adalah kebenaran bahwa citra wajah yang diujikan benar-benar citra wajah orang yang dimaksud.

7. Metode *Principal Component Analysis* (PCA)

Principal Component Analysis (PCA) adalah sebuah cara untuk mengidentifikasi pola pada data dan kemudian mengekspresikan data tersebut ke bentuk yang lain untuk menunjukkan perbedaan dan persamaan antar pola. Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (*observed variables*) menjadi dimensi yang lebih kecil dari ruang fitur (*independent variables*), yang dibutuhkan untuk mendeskripsikan data lebih sederhana. Ruang fitur adalah ciri yang digunakan sebagai kriteria dalam pengklasifikasian (Pertiwi & Harjoko, 2013). *Principal Component Analysis* menggunakan vektor-vektor yang disebut dengan *eigenvector* dan nilai-nilai yang disebut dengan *eigenvalue* untuk mendapatkan fitur yang paling signifikan pada dataset. Prinsip dasar dari algoritma *Principal Component Analysis* adalah mengurangi satu set data namun tetap mempertahankan sebanyak mungkin variasi dalam set data tersebut. Secara matematis *Principal Component Analysis* mentransformasikan sebuah variabel yang

berkolerasi ke dalam bentuk yang bebas tidak berkolerasi.

Algoritma *Principal Component Analysis* (PCA) merupakan salah satu metode yang dapat digunakan untuk mengolah citra wajah seseorang sehingga secara otomatis sistem akan mengenali wajah seseorang melalui ciri-ciri utamanya seperti mata, hidung, bibir, alis sebagai identitas. Identitas dari citra wajah seseorang tersebut oleh sistem akan dikenali melalui berbagai pelatihan (*training*) yang disimpan di database. Fase pelatihan (*training*) merupakan hasil ekstraksi dari kumpulan berbagai wajah yang berbeda kemudian dikumpulkan dan disimpan disebuah database. Hasil citra wajah yang telah diekstraksi menggunakan algoritma PCA tersebut nantinya akan dibandingkan dengan citra wajah baru sebagai citra wajah yang akan dites apakah mempunyai kemiripan atau hampir mirip untuk dikenali oleh sistem.

Dalam *pattern recognition*, fitur-fitur suatu citra menggunakan PCA direpresentasikan sebagai *eigenvectors*. *Eigenvectors* didefinisikan sebagai kumpulan hubungan karakteristik-karakteristik dari suatu citra untuk mengenali citra tersebut secara spesifik.

8. Algoritma *Principal Component Analysis* (PCA)

Algoritma pengenalan wajah menggunakan *Principal Component Analysis* dimulai dengan membuat matriks kolom dari wajah yang di-input ke dalam database. Matriks kolom tersebut akan diubah menjadi bentuk *flatvector* untuk dicari Rata-rata *vector* citra dari gambar wajah atau disebut dengan Rataan *FlatVector*. Rataan *FlatVector* dihitung dengan cara membagi penjumlahan dari seluruh *flatvector* citra dengan jumlah banyaknya citra yang disimpan di dalam database.

9. Penyusunan *Flatvector* Matriks Citra

Langkah pertama adalah menyusun seluruh *training image* menjadi 1 matriks

tunggal. Misalnya *image* yang kita simpan berukuran $H \times W$ *pixel* dan jumlahnya N buah, maka memiliki *flatvector* dengan dimensi $N \times (H \times W)$. Representasikan semua matriks *training* menjadi matriks dengan bentuk $N \times 1$ atau matriks linier seperti yang ditunjukkan pada gambar 2.1.

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \rightarrow [a \ b \ c \ d \ e \ f]$$

Gambar 2.1 Skema Matrik Wajah

Contoh dibawah ini menggunakan empat wajah citra yang telah diubah menjadi matriks, lalu matriks tersebut diubah kedalam bentuk rata-rata *FlatVector*.

$$\begin{aligned} C_1 &\rightarrow \begin{bmatrix} 11 & 11 & 11 \\ 11 & 11 & 11 \\ 11 & 11 & 11 \end{bmatrix} \rightarrow [11 \ 11 \ 11 \ 11 \ 11 \ 11] \\ C_2 &\rightarrow \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow [2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2] \\ C_3 &\rightarrow \begin{bmatrix} 9 & 9 & 9 \\ 9 & 9 & 9 \\ 9 & 9 & 9 \end{bmatrix} \rightarrow [9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9] \\ C_4 &\rightarrow \begin{bmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix} \rightarrow [8 \ 8 \ 8 \ 8 \ 8 \ 8 \ 8 \ 8 \ 8] \end{aligned}$$

10. Hitung Rata-rata *Flatvector*

Dari *FlatVector* yang diperoleh, jumlahkan seluruh barisnya dan bagi dengan jumlah *image training* untuk mendapatkan Rata-rata (*mean*) *FlatVector*.

$$\begin{aligned} \text{Rata-rata flatvector} &= \frac{C_1 + C_2 + \dots + C_n}{n} \\ \text{Rataan flatvector} &= \frac{C_1 + C_2 + C_3 + C_4}{4} \\ &= \begin{bmatrix} 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \end{bmatrix} \\ &= \frac{30 + 30 + 30 + 30 + 30 + 30 + 30 + 30 + 30}{4} \\ &= [7.5 \ 7.5 \ 7.5 \ 7.5 \ 7.5 \ 7.5 \ 7.5 \ 7.5 \ 7.5] \end{aligned}$$

11. Tentukan Nilai *Eigenface*

Dengan menghitung rata-rata *flatvector* citra, maka nilai *eigenface* untuk matriks *flatvector* yang sudah disusun tersebut dapat dihitung nilai *eigenface*-nya. Caranya dengan

mengurangi baris- baris pada matriks *flatvector* dengan rata-rata *flatvector*. Jika didapatkan nilai di bawah nol (nilai minus), maka nilainya diganti dengan nol.

$$C_1 = \frac{\begin{matrix} 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{matrix}}{\begin{matrix} 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}} -$$

$$C_2 = \frac{\begin{matrix} 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\ 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 \end{matrix}}{\begin{matrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{matrix}} -$$

2.2.1.1 Proses Identifikasi

Untuk mengenali citra tes (*testface*) pada saat uji, langkah identifikasinya adalah hitung nilai *eigenface* untuk matriks *testface* dengan cara yang sama seperti sebelumnya yaitu dimulai dari awal penentuan nilai *flatvector*, dikurangi dengan rata-rata *FlatVector* (didapat dari citra *training*), dan mendapatkan *eigenface* untuk *testface*

$$Ct = \begin{bmatrix} 8 & 9 & 9 \\ 9 & 6 & 9 \\ 9 & 9 & 7 \end{bmatrix} \rightarrow [8 \ 9 \ 9 \ 9 \ 6 \ 9 \ 9 \ 9 \ 7]$$

$$= \left[\begin{matrix} 8 & 9 & 9 & 9 & 6 & 9 & 9 & 9 & 7 \\ 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 0 & 1.5 & 1.5 & 1.5 & 0 \end{matrix} \right]$$

Identifikasi dilakukan dengan metode *Euclidean Distance* yaitu menentukan jarak (*distance*) terpendek antara nilai *eigenface* dari *training image* di *database* dengan *eigenface* dari *testface*. Proses dimulai dengan menentukan nilai absolut dari pengurangan matriks *eigenface* *training image* dengan *eigenface* dari *testface* dan jumlahkan seluruh elemen penyusun

vector yang dihasilkan. Kemudian cari nilai paling kecil dari hasil penjumlahan tersebut.

$$|identify Cn| = \frac{\text{Eigenface Training}}{\text{Eigenface Testface}} \frac{a \ b \ c \ d \ f \ g \ h \ i \ j \ \dots}{\dots}$$

$$|C1| = \frac{\begin{matrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 0 & 1.5 & 1.5 & 1.5 & 0 \end{matrix}}{\begin{matrix} 1 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 & 1.5 \end{matrix}} -$$

$$= 1 + 0 + 0 + 0 + 1.5 + 0 + 0 + 0 + 1.5$$

$$= 4$$

$$|C2| = \frac{\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 1.5 & 1.5 & 1.5 & 0 & 1.5 & 1.5 & 1.5 & 0 \end{matrix}}{\begin{matrix} 0.5 & 1.5 & 1.5 & 1.5 & 0 & 1.5 & 1.5 & 1.5 & 0 \end{matrix}} -$$

$$= 0.5 + 1.5 + 1.5 + 1.5 + 0 + 1.5 + 1.5 + 1.5 + 0$$

$$= 9.5$$

$$|C3| = \frac{\begin{matrix} 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 0 & 1.5 & 1.5 & 1.5 & 0 \end{matrix}}{\begin{matrix} 3 & 2 & 2 & 2 & 3.5 & 2 & 2 & 2 & 3.5 \end{matrix}} -$$

$$= 3 + 2 + 2 + 2 + 3.5 + 2 + 2 + 2 + 3.5$$

$$= 22$$

$$|C4| = \frac{\begin{matrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 0 & 1.5 & 1.5 & 1.5 & 0 \end{matrix}}{\begin{matrix} 0 & 1 & 1 & 1 & 0.5 & 1 & 1 & 1 & 0.5 \end{matrix}} -$$

$$= 0 + 1 + 1 + 1 + 0.5 + 1 + 1 + 1 + 0.5$$

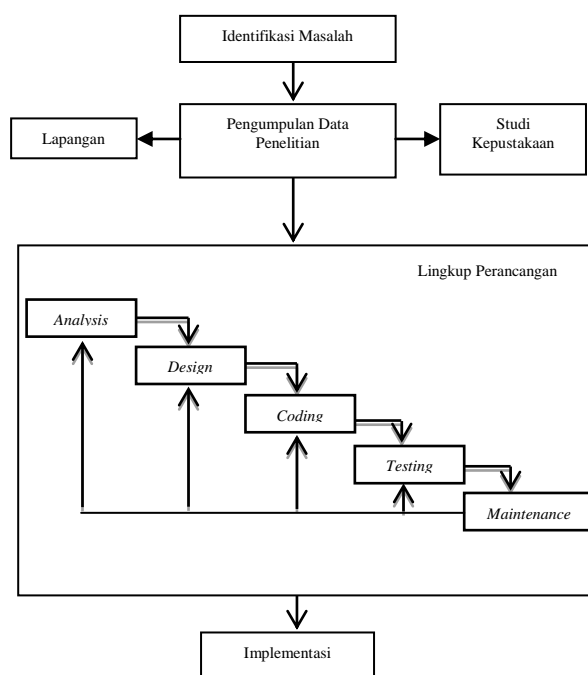
$$= 7$$

Dari hasil perhitungan diperoleh jarak citra wajah satu memiliki nilai yang terkecil yaitu 4. Karena jarak *eigenface face* satu dengan *eigenface testface* paling kecil, maka hasil identifikasi menyimpulkan bahwa *testface* lebih mirip dengan *face* satu dari pada *face* dua, *face* tiga, dan *face* empat.

III. METODE

1. Kerangka Penelitian

Kerangka penelitian menjelaskan tentang langkah dan tahapan-tahapan yang dilakukan selama penelitian untuk memudahkan penulis dalam mengerjakan tugas akhir ini.



Gambar 3.1 Kerangka Penelitian

2. Identifikasi Masalah

Identifikasi masalah adalah salah satu proses penelitian yang boleh dikatakan penting diantara proses lain. Dalam tahap ini penulis mencoba untuk mengidentifikasi permasalahan yang ada. Tahap ini menjelaskan tahap awal pada penyusunan penelitian ini. Hasil dari identifikasi inilah yang menjadi objek penelitian. Masalah yang diidentifikasi adalah bagaimana merancang suatu aplikasi pengenalan wajah (*face recognition*) dengan menggunakan metode *Principal Componen Analysis*.

3. Pengumpulan Data Penelitian

Untuk mendapatkan data yang mendukung penelitian ini, penulis menggunakan metode yaitu:

a. Studi Pustaka

Metode pengumpulan data dalam penulisan ini menggunakan studi kepustakaan guna menunjang pembuatan tugas akhir dengan membaca dan mempelajari buku-buku literatur, jurnal, artikel yang berhubungan dengan

pengenalan citra berbasis komputer khususnya pengenalan wajah manusia.

b. Survei

Survei yang dilakukan dalam penelitian ini adalah melakukan pengumpulan citra wajah yang digunakan sebagai sampel untuk database dan uji coba dalam aplikasi ini. Adapun sampel sebanyak 10 orang dengan kriteria sebagai berikut :

Tabel 3.1 Kriteria Sampel

Pria	Wanita	Ekspresi
Pakai Kacamata	Berjilbab	Senyum
Tidak Pakai Kacamata	Berjilbab pakai Kacamata	Normal
Berkumis		Ketawa

4. Kerangka Pengembangan Aplikasi

Kerangka pengembangan aplikasi yang digunakan adalah *sequential linier (waterfall)*. Penjelasan tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut :

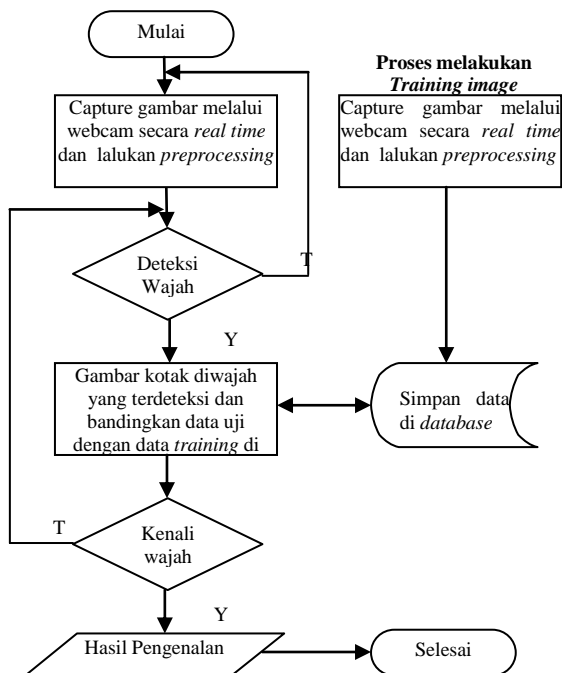
1. *Analysis*. Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.
2. *Design*. Tahap penerjemahan dari data yang dianalisis kedalam bentuk yang mudah dimengerti.
3. *Coding*. Tahap penerjemahan data atau pemecahan masalah yang telah dirancang kedalam bahasa pemrograman C#.
4. *Testing*. Merupakan tahap pengujian terhadap perangkat lunak yang dibangun.
5. *Maintenance*. Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan-perubahan atau penambahan sesuai dengan kebutuhan.

IV. HASIL PEMBAHASAN

pengenalan wajah dalam keadaan yang telah dikondisikan sebelumnya.

Berdasarkan kebutuhan sistem dan hasil analisis, maka sistem yang akan dibuat harus dapat memenuhi fungsi-fungsi sebagai berikut:

1. Mengimplementasikan metode *Principal Componen Analysis* pada sistem pengenalan wajah.
2. Menguji performa dan kemampuan metode *Principal Componen Analysis* pada sistem pengenalan wajah dalam keadaan yang telah dikondisikan sebelumnya.
3. Menghasilkan sebuah kesimpulan berupa tingkat akurasi metode *Principal Componen Analysis* dalam sistem pengenalan wajah.

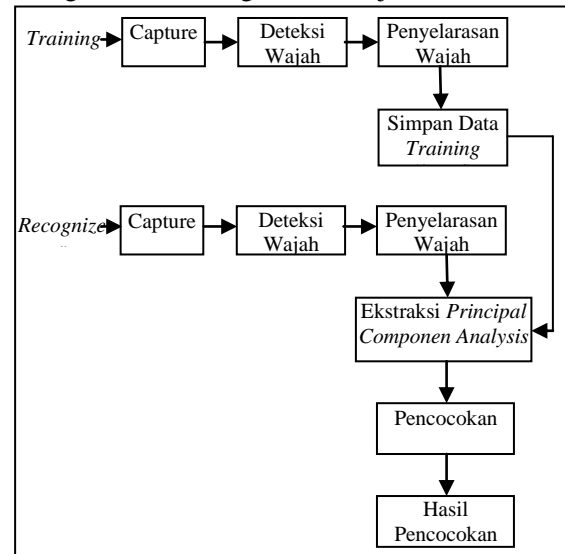


Gambar 4.1 Flowchart Sistem

1. Perancangan Sistem

Proses dari sistem pengenalan wajah menggunakan metode *Principal Componen Analysis* ini terdapat beberapa komponen yang dapat digambarkan dalam blok diagram. Setelah itu akan dilakukan proses perancangan aplikasi.

Diagram Blok Pengenalan Wajah

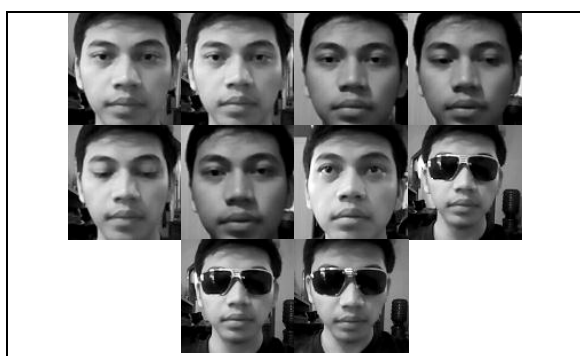


Gambar 4.2 Diagram Blok Pengenalan Wajah

Berdasarkan pada gambar 4.1 diatas dapat dijelaskan bahwa komponen *training* merupakan data wajah yang akan dilatih dan disimpan dalam *database* yang nantinya akan digunakan untuk proses pencocokan wajah dengan data wajah yang diuji, sedangkan komponen *recognizer* merupakan data wajah yang akan diuji dan dicocokkan dengan data wajah yang ada di dalam *database*, yang mana jika ada kemiripan dengan data uji dengan data latih, maka hasil pencocokan akan ditampilkan.

Berikut contoh sampel data *training* dari sistem pengenalan wajah menggunakan metode PCA :





Gambar 4.3 Contoh Data Sampel

Penjelasan mengenai proses pengenalan wajah menggunakan PCA, Untuk lebih jelasnya algoritma PCA dapat diuraikan dalam ilustrasi sebagai berikut :

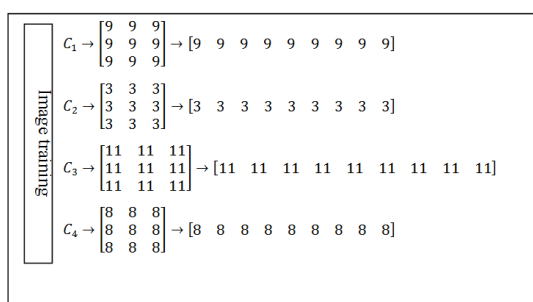
2. Penyusunan *Flatvector* matriks citra

Langkah pertama adalah menyusun seluruh *training image* menjadi 1 matriks tunggal. Misalnya *image* yang kita simpan berukuran $H \times W$ piksel dan jumlahnya N buah, maka memiliki *flatvector* dengan dimensi $N \times (H \times W)$. representasikan semua matriks *training* menjadi matriks dengan bentuk $N \times 1$ atau matriks linier seperti yang ditunjukkan berikut ini:

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \rightarrow [a \ b \ c \ d \ e \ f]$$

Gambar 4.4 Skema Matriks Wajah

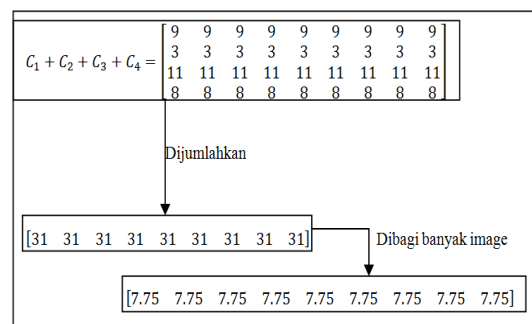
Misalnya didalam *training image* terdapat tiga *image* ukuran 3×3 piksel maka kita akan mempunyai *eigenvector* ukuran 2×9 . Contoh dibawah ini menggunakan empat wajah citra yang telah diubah menjadi matriks, lalu matriks tersebut diubah kedalam bentuk rataaan *FlatVector*.



Gambar 4.5 Skema *FlatVector* Wajah

1. Hitung Rataan *FlatVector*

Dari *FlatVector* yang diperoleh, jumlahkan seluruh barisnya sehingga kita peroleh matriks berukuran $1 \times (H \times W)$.

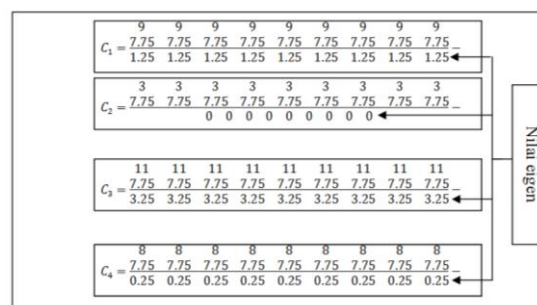


Gambar 4.6 Rataan *FlatVector*

Nilai *flatvector* citra akan digunakan untuk menghitung nilai *eigenface* citra wajah untuk *training image*.

2. Tentukan nilai *eigenface*

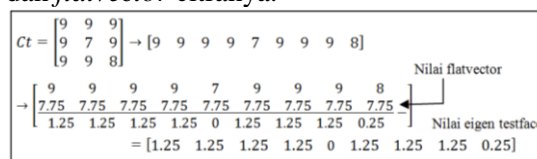
Dengan memakai rataaan *flatvector* citra di atas nilai *eigenface* untuk matriks *flatvector* yang sudah disusun tersebut dapat dihitung nilai *eigenfacenya*. Caranya dengan mengurangi baris-baris pada matrix *flatvector* dengan rataaan *flatvector*. Jika didapatkan nilai dibawah nol, maka nilainya diganti dengan nol.



Gambar 4.7 Hitung Nilai Eigen Citra Training

3. Proses Identifikasi

Untuk mengenali citra tes (*testface*), langkah identifikasinya adalah hitung nilai *eigenface* untuk matriks *testface* dengan cara sebelumnya untuk penentuan nilai *eigenface* dan *flatvector* citranya.



Gambar 4.8 Nilai Eigen Image Test

Nilai eigen (*eigenvalue*) dari *testface* digunakan untuk identifikasi dengan menentukan jarak terpendek dengan *eigenface* dari *eigenvector training* dengan cara menentukan nilai *absolut* dari pengurangan baris I pada matriks *eigenface training* citra dengan *eigenface* dari *testface* dan jumlahkan dengan elemen penyusun *vector* yang dihasilkan dari pengurangan dan didapat jarak indeks I dan cari nilai yang paling kecil. Setelah nilai *eigenface* untuk *testface* diperoleh maka kita bisa melakukan identifikasi dengan menentukan jarak terpendek dengan *eigenface* dari *eigenvector training image*. Pertama tentukan nilai *absolut* dari pengurangan baris pada matriks *eigenface training image* dengan *eigenface* dari *testface*, kemudian jumlahkan elemen-elemen penyusun *vector* yang dihasilkan dari pengurangan tadi dan ditemukan jarak indeks. Lakukan untuk semua baris. Cari nilai yang paling kecil.

1. Nilai Eigenface $C_1 = (1.25 \ 1.25 \ 1.25 \ 1.25 \ 1.25 \ 1.25 \ 1.25 \ 1.25 \ 1.25 \ 1.25)$

$$\begin{array}{r} C1 \rightarrow \begin{array}{cccccccccc} 1.25 & 1.25 & 1.25 & 1.25 & 1.25 & 1.25 & 1.25 & 1.25 & 1.25 & 1.25 \end{array} \\ C_t \rightarrow \begin{array}{cccccccccc} 1.25 & 1.25 & 1.25 & 1.25 & 0 & 1.25 & 1.25 & 1.25 & 0.25 & 0.25 \end{array} \\ \hline \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 1.25 & 0 & 0 & 0 & 1 & 1 \end{array} \end{array}$$

Nilai jarak :

$$= 0+0+0+0+1.25+0+0+1$$

$$= 2.25$$

2. Nilai Eigenface $C_2 = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$

$$\begin{array}{r} C2 \rightarrow \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \\ C_t \rightarrow \begin{array}{cccccccccc} 1.25 & 1.25 & 1.25 & 1.25 & 0 & 1.25 & 1.25 & 1.25 & 0.25 & 0.25 \end{array} \\ \hline \begin{array}{cccccccccc} -1.25 & -1.25 & -1.25 & -1.25 & 0 & -1.25 & -1.25 & -1.25 & 0.25 & 0.25 \end{array} \end{array}$$

Nilai jarak :

$$= 1.25+1.25+1.25+1.25+1.25+1.25+1.25+1$$

$$= 11$$

3. Nilai Eigenface $C_3 = (3.25 \ 3.25 \ 3.25 \ 3.25 \ 3.25 \ 3.25 \ 3.25 \ 3.25 \ 3.25 \ 3.25)$

$$\begin{array}{r} C3 \rightarrow \begin{array}{cccccccccc} 3.25 & 3.25 & 3.25 & 3.25 & 3.25 & 3.25 & 3.25 & 3.25 & 3.25 & 3.25 \end{array} \\ C_t \rightarrow \begin{array}{cccccccccc} 1.25 & 1.25 & 1.25 & 1.25 & 0 & 1.25 & 1.25 & 1.25 & 0.25 & 0.25 \end{array} \\ \hline \begin{array}{cccccccccc} 2 & 2 & 2 & 2 & 3.25 & 2 & 2 & 2 & 3 & 3 \end{array} \end{array}$$

Nilai jarak :

$$= 2+2+2+2+3.25+2+2+2+3$$

$$= 18.25$$

4. Nilai Eigenface $C_4 = (0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25)$

$$\begin{array}{r} C1 \rightarrow \begin{array}{cccccccccc} 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \end{array} \\ C_t \rightarrow \begin{array}{cccccccccc} 1.25 & 1.25 & 1.25 & 1.25 & 0 & 1.25 & 1.25 & 1.25 & 0.25 & 0.25 \end{array} \\ \hline \begin{array}{cccccccccc} -1 & -1 & -1 & -1 & 0.25 & -1 & -1 & -1 & -1 & -1 \end{array} \end{array}$$

Nilai jarak :

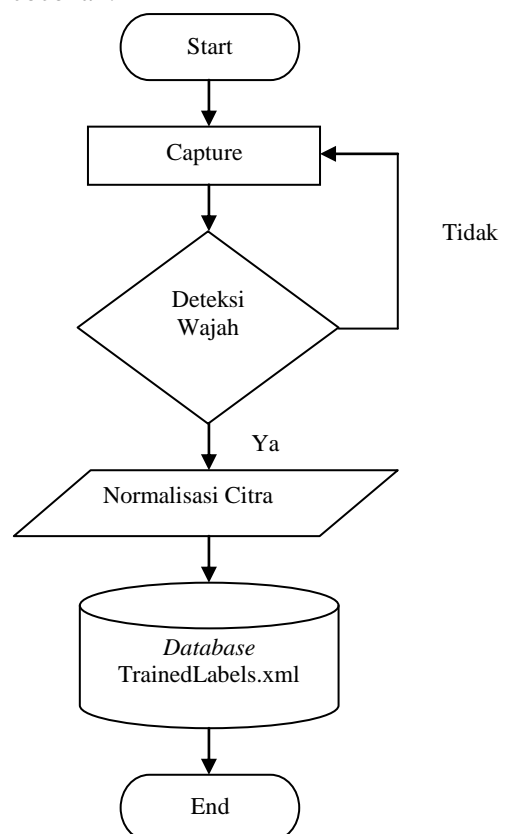
$$= 1+1+1+1+0.25+1+1+1+1$$

$$= 8.25$$

Dari hasil perhitungan, diperoleh jarak citra wajah citra satu memiliki nilai yang terkecil yaitu 2.25. Karena jarak *eigenface* satu dengan *testface* yang paling kecil, maka hasil identifikasi menyimpulkan bahwa *testface* lebih mirip dengan *face* satu dari pada *face* dua, *face* tiga dan *face* empat.

4. Flowchart Training Wajah

Pada proses ini akan dilakukan pembelajaran terhadap citra yang akan disimpan ke dalam database. Hasil dari pembelajaran ini kemudian disimpan dengan format *.xml dan digunakan untuk proses pencocokan.



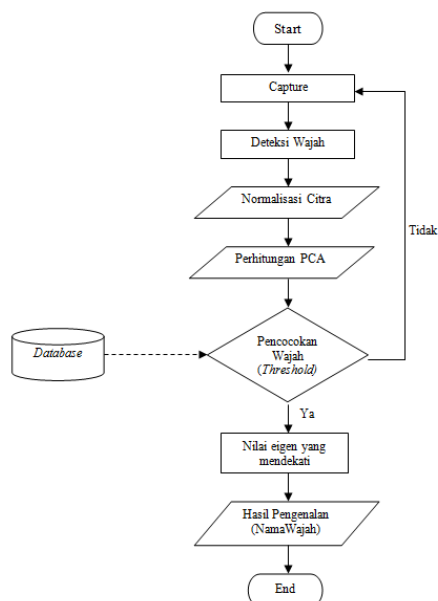
Gambar 4.9 Flowchart Training Wajah

Dari gambar diatas, langkah *Training* wajah dapat dijelaskan sebagai berikut :

1. Mulai dari *start*, input dimulai dengan mengambil *capture* dari *webcam*
2. Setelah melakukan *capture*, langkah selanjutnya mendeteksi wajah. Jika wajah tidak terdeteksi kembali ke proses *capture*. Sedangkan jika terdeteksi, akan berlanjut.
3. Setelah wajah terdeteksi, langkah selanjutnya adalah proses normalisasi citra. Normalisasi tersebut meliputi *resizing* citra 100x100 pixel dan setelahnya dilakukan perubahan format gambar dari RGB ke *grayscale*.
4. Setelah proses normalisasi citra, selanjutnya adalah data *Training* akan disimpan kedalam *database* yang berformat *.xml.
5. Proses selesai.

5. Flowchart Sistem Pengenalan Wajah

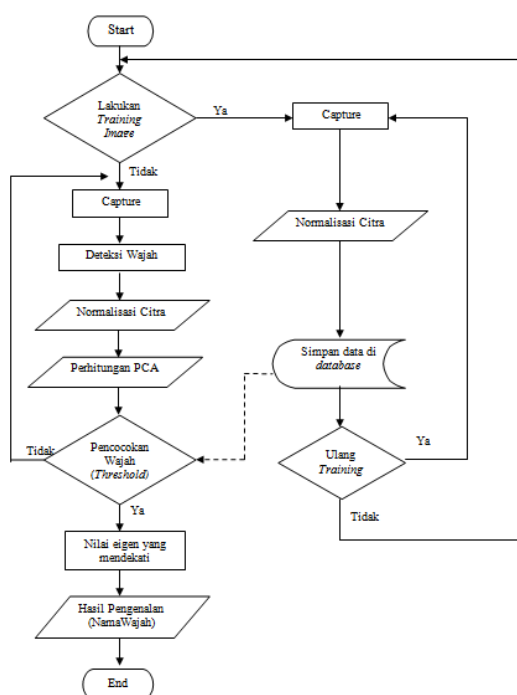
Berikut perancangan *flowchart* dari sistem pengenalan wajah menggunakan metode *Principal Componen Analysis*, dan metode *euclidean distance* untuk proses pencocokan wajah.



Gambar 4.10 Flowchart Proses Pengenalan Wajah

Dari gambar diatas, langkah pengenalan wajah dapat dijelaskan sebagai berikut :

1. Mulai dari *start*, input dimulai dengan mengambil *capture* dari *webcam*
2. Setelah melakukan *capture*, langkah selanjutnya mendeteksi wajah.
3. Setelah wajah terdeteksi, langkah selanjutnya adalah proses normalisasi citra. Normalisasi tersebut meliputi *resizing* citra 100x100 pixel dan setelahnya dilakukan perubahan format gambar dari RGB ke *grayscale*, kemudian dari *grayscale* tersebut diubah lagi menjadi bentuk matriks.
4. Setelah proses normalisasi citra, selanjutnya adalah proses perhitungan *Principal Componen Analysis* data citra uji dengan data citra yang ada di *database* dengan ketentuan data wajah yang akan dikenali diatas nilai *Threshold*.
5. Tahap selanjutnya adalah pencocokan nilai PCA input dengan PCA data *training (database)* menggunakan metode *euclidean distance*, jika tidak cocok kembali ke proses *capture*. Sedangkan jika cocok, akan berlanjut.
6. Selanjutnya jika cocok, akan dicari nilai eigen yang mendekati.
7. Setelah ditemukan kemiripan (nilai eigen yang mendekati) dengan citra wajah, hasil nama pengenalan wajah akan ditampilkan.
8. Proses selesai.



Gambar 4.11 Flowchart Aplikasi

6. Implementasi

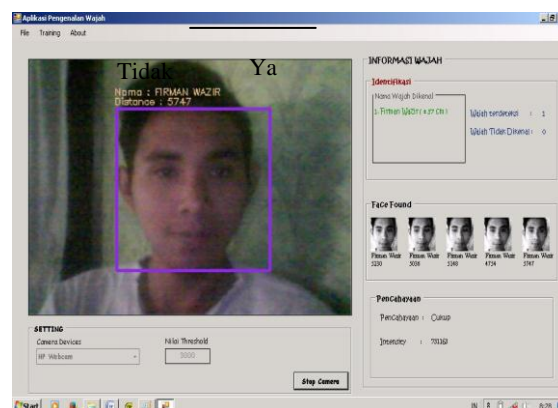
Implementasi merupakan tahapan setelah melakukan analisis dan perancangan sistem pada siklus rekayasa perangkat lunak, dimana aplikasi siap untuk dioperasikan pada keadaan yang sebenarnya sehingga akan diketahui apakah program yang telah dibuat benar-benar dapat menghasilkan sebuah keluaran yang sesuai dengan tujuan yang diinginkan.

7. Desain User Interface

Pembuatan *User Interface* pengenalan wajah menggunakan sarana yang diberikan oleh *Visual Studio 2010*, *User Interface* berguna untuk mempermudah pengguna didalam penginputan data yang dibutuhkan untuk sistem.

8. Tampilan Form Utama

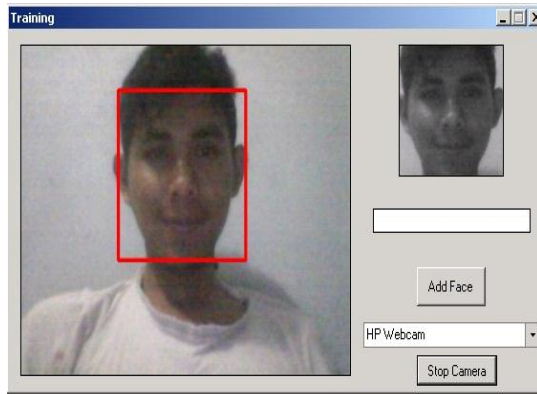
Form utama merupakan tampilan awal dari aplikasi pengenalan wajah, yang mana dalam form utama ini terdapat 3 *menustrip*, 1 *picturebox*, dan beberapa komponen lainnya. Gambar 4.6 dibawah ini merupakan tampilan dari form utama.



Gambar 4.12 Tampilan Form Utama

Form utama ini guna untuk proses pengenalan wajah, yang mana komponen *picturebox* merupakan sebagai *webcam*. Pada komponen *groupbox setting* terdapat 1 *combobox*, 1 *Textbox* dan 1 *button*, komponen *combobox* digunakan untuk memilih *devices camera* yang akan digunakan, sedangkan komponen *Textbox* digunakan untuk mengatur nilai *threshold* (nilai ambang batas), nilai *threshold* ini untuk menentukan batas nilai pencocokan wajah yang jika nilai jarak kemiripan berada diatas *threshold*, maka wajah dikenali dan jika dibawah *threshold* wajah tidak dikenali. Sedangkan Tombol *button* pada komponen *groupbox setting* digunakan untuk yang jika di-klik maka *capture* dari *webcam* akan ditampilkan sesuai *devices camera* yang telah dipilih. Komponen *groupbox informasi wajah* guna sebagai informasi dari wajah yang dideteksi, baik wajah yang dikenali atau yang tidak dikenali, komponen ini akan menampilkan informasi sesuai hasil dari proses pengenala wajah.

Dalam form utama juga terdapat beberapa komponen *menustrip*, diantaranya *menustrip training*, menu ini digunakan untuk masuk ke form proses *training image* (proses pelatihan wajah), yang mana proses *training* untuk pembelajaran citra wajah yang nantinya digunakan untuk proses pencocokan wajah pada form utama proses pengenalan wajah.



Gambar 4.13 Tampilan Form Training

Gambar 4.7 merupakan form untuk melakukan *training* citra wajah, form *training* ini guna untuk melakukan pembelajaran pada citra wajah dan hasil pembelajaran itu akan disimpan kedalam *database* yang berformat *.xml untuk dijadikan sampel. Data sampel ini yang akan digunakan untuk proses pencocokan wajah dengan data wajah uji.




Gambar 4.14 Tampilan Form Profile

9. Pengujian Sistem Pengenalan Wajah

Pengujian ini dimaksudkan untuk mengetahui akurasi sistem pengenalan wajah dengan menggunakan jumlah data *training* dan data yang diuji yang berbeda. Berikut hasil untuk pengujian pengenalan wajah :

Tabel 4.1 Hasil Pengujian Pertama

Data Wajah Yang Diuji	Jenis Kelamin	Atribut	Ekspr esi	Hasil	Presentase Keberhasilan
 Firman Wazir	Laki-Laki	Kacamata	Senyum	Diken ali	$\frac{6}{6} * 100\% = 100\%$
		Kacamata	Normal	Diken ali	
		Kacamata	Sedih	Diken ali	
		Tanpa Kacamata	Senyum	Diken ali	
		Tanpa Kacamata	Normal	Diken ali	
		Tanpa Kacamata	Sedih	Diken ali	

Berdasarkan hasil pengujian terhadap sampel data wajah pertama yang ditunjukkan pada tabel 4.1, dilakukan pengujian sebanyak

6 kali. Hasil pengujian menunjukkan presentase tingkat keberhasilan (akurasi) yaitu 100%.

Tabel 4.2 Hasil Pengujian Kedua

Data Wajah Yang Diuji	Jenis Kelamin	Atribut	Ekspresi	Hasil	Presentase Keberhasilan
 Tony	Laki-Laki	Kacamata	Senyum	Dikenali	$6/6 * 100\% = 100\%$
		Kacamata	Normal	Dikenali	
		Kacamata	Sedih	Dikenali	
		Tanpa Kacamata	Senyum	Dikenali	
		Tanpa Kacamata	Normal	Dikenali	
		Tanpa Kacamata	Sedih	Dikenali	

Hasil pengujian terhadap sampel data wajah kedua yang ditunjukan pada tabel 4.2, dilakukan pengujian sebanyak 6 kali. Hasil

pengujian menunjukkan presentase tingkat keberhasilan (akurasi) sama dengan hasil pengujian sampel yang pertama yaitu 100%.


Tabel 4.3 Hasil Pengujian Ke Tiga

Data Wajah Yang Diuji	Jenis Kelamin	Ekspresi	Hasil	Presentase Keberhasilan
 Rizki	Laki-Laki	Senyum	Dikenali	$5/6 * 100\% = 83\%$
		Sedih	Dikenali	
		Normal	Dikenali	
		Normal	Dikenali	
		Sedih	Tidak	
		Senyum	Dikenali	

Hasil pengujian terhadap sampel wajah ketiga pada tabel 4.3, dilakukan pengujian sebanyak 6 kali, Hasil pengujian

menunjukkan presentase tingkat keberhasilan (akurasi) yaitu 83%.

Tabel 4.4 Hasil Pengujian Ke Empat

Data Wajah Yang Diuji	Jenis Kelamin	Atribut	Ekspresi	Hasil	Presentase Keberhasilan
 Ibu Tony	Perempuan	Kacamata, Jilbab	Senyum	Tidak	$6/10 * 100\% = 60\%$
		Kacamata, Jilbab	Sedih	Tidak	
		Kacamata, Jilbab	Senyum	Dikenali	

		Kacamata, Jilbab	Normal	Dikenali	
		Kacamata, Jilbab	Sedih	Dikenali	
		Jilbab	Senyum	Tidak	
		Jilbab	Senyum	Dikenali	
		Jilbab	Normal	Dikenali	
		Jilbab	Sedih	Tidak	
		Jilbab	Sedih	Dikenali	

Hasil pengujian terhadap sampel data wajah keempat pada tabel 4.4, dilakukan pengujian sebanyak 10 kali, karena pengaruh pencahayaan yang membuat wajah kadang dikenali dan tidak dikenali. Hasil pengujian menunjukkan presentase tingkat keberhasilan (akurasi) yaitu 60%.

Tabel 4.5 Hasil Pengujian Ke Lima

Data Wajah Yang Diuji	Jenis Kelamin	Cahaya	Ekspresi	Hasil	Presentase Keberhasilan
 Irham	Laki-Laki	Terang	Senyum	Dikenali	$8/10 * 100\%$ =80%
		Terang	Sedih	Dikenali	
		Terang	Normal	Dikenali	
		Cukup	Normal	Dikenali	
		Cukup	Sedih	Dikenali	
		Cukup	Senyum	Dikenali	
		Kurang	Senyum	Tidak	
		Kurang	Normal	Dikenali	
		Kurang	Sedih	Tidak	
		Kurang	Sedih	Dikenali	

Hasil pengujian terhadap sampel wajah kelima pada tabel 4.5, dilakukan pengujian sebanyak 10 kali, Hasil pengujian menunjukkan presentase tingkat keberhasilan (akurasi) yaitu 80%. Pengujian ini menunjukkan bahwa pencahayaan untuk akurasi pengenalan wajah sangat mempengaruhi. Walau pada tabel diatas akurasi pengenalan wajah menunjukkan lumayan bagus, tapi sewaktu-sewaktu jika

diuji kembali hasilnya bisa berbeda, itu karena pencahayaan dan juga kualitas citra sampel yang diambil tidak sama rata dengan citra sampel yang lainnya.

Berdasarkan pengujian yang dilakukan, tingkat keberhasilan pengenalan wajah menggunakan metode *principal component analysis* dengan 38 kali pengujian adalah 81% sukses dikenali dan 29% gagal dikenali. Penyebab kurangnya keakuratan

dalam pengenalan wajah menggunakan metode *principal component analysis* ini adalah :

1. Sensitif terhadap cahaya
2. Jarak wajah dari kamera terlalu jauh
3. Perubahan wajah yang terlalu ekstrim Ekspresi wajah

V. KESIMPULAN

Pada bab ini dijelaskan kesimpulan yang diambil berdasarkan analisa dan saran yang diambil berdasarkan hasil pengujian yang telah dilakukan.

Kesimpulan dari Tugas Akhir ini berdasarkan pengujina, hasil dan analisa adalah :

1. Pengenalan pola wajah dapat dikenali menggunakan nilai *pixel* dari data wajah.
2. Proses pengenalan wajah menggunakan pendekatan metode *Principal Componen Analysis* sensitif terhadap perubahan cahaya, jarak, ekspresi wajah, sudut pandang wajah dan perubahan wajah yang terlalu ekstrim. Jika citra yang digunakan sebagai *training* set maupun sebagai citra input memiliki intensitas cahaya yang berbeda dan tidak berada pada posisi yang sama dengan citra *training* set maka proses tersebut tidak dapat memberikan hasil yang akurat
3. metode *Principal Componen Analysis* dapat diimplementasikan untuk pengenalan wajah dengan tingkat akurasi 81%.
4. Wajah dapat dikenali apabila nilai jarak antara *image training* dengan *image test* diatas nilai *threshold*.

DAFTAR PUSTAKA

- Damayanti, Fitri, Agus Zainal Abidin dan Rully Soelaiman. 2010. *Pengenalan Citra Wajah Menggunakan Metode Two-Dimensional Linear Discriminant Analysis dan Support Vector Machine*. Jurnal Ilmiah Kursor, Vol.5, No.3
- Fatta, Hanif Al. 2009. *Rekayasa Sistem Pengenalan Wajah*. Andi: Yogyakarta.
- Ismawan, Fiqih. 2015. *Hasil Ekstraksi Algoritma Principal Component Analysis (PCA) Untuk Pengenalan Wajah Dengan Bahasa Pemrograman Java Eclipse IDE*. Sisfotek Global, Vol.5, No.1
- Lestari, Sri Puji. 2014. *Implementasi Pengenalan Wajah Manusia Menggunakan Principal Component Analysis (PCA)*. Pelita Informatika Budi Darma, Vol.6, No. 2.
- Marti, Ni Wayan. 2010. *Pemanfaatan gui dalam pengembangan perangkat lunak Pengenalan citra wajah manusia menggunakan metode Eigenfaces*. Seminar Nasional Aplikasi Teknologi Informasi 2010: Yogyakarta
- Pratiwi, Dian Esti dan Agus Harjoko. 2013. *Implementasi Pengenalan Wajah menggunakan PCA (Principal Component Analysis)*. Ijeis, Vol.3, No.2
- Rejeki, Muslim Setyo dan Ali Tarmuji. 2013. *Membangun aplikasi autogenerate script ke Flowchart untuk mendukung business process Reengineering*. Jurnal Sarjana Teknik Informatika, Vol.1, No.2
- Sembiring, Maha Kinnoy. 2015. *Perancangan Aplikasi Pengenalan Wajah Menggunakan Metode Fisher Linear Discriminant (FLD)*. Pelita Informatika Budi Darma, Vol.9, No.1
- Suryanto, Agus dan Samiyono. 2011. *Implementasi Model Analisis*

- Perbaikan Faktor Daya Listrik Rumah Tangga dengan Simulasi Perangkat Lunak.* Jurnal Kompetensi Teknik, Vol.3, No.1
- Indra. 2012. *Sistem Pengenalan Wajah Dengan Metode Eigenface Untuk Absensi Pada PT. Florindo Lestari.* Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012. ISBN 979-26-0255-0
- Muliawan, Muhammad Rizki, Beni Irawan dan Yulrio Brianorman. *Implementasi Pengenalan Wajah Dengan Metode Eigenface Pada Sistem Absensi.* Jurnal Coding. Vol. 3, No. 1