

**APLIKASI PENGENALAN WAJAH BERBASIS
*OPENCV***

TUGAS AKHIR

Oleh :

Swara Adi Cakti Prima Dasa	3310801114
Fitra Fajrianto	3310801116

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2011**

LEMBAR PENGESAHAN

Batam, 22 Agustus 2011

Pembimbing,

Riwinoto, M.Kom

NIK. 103025

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801114

Nama : Swara Adi Cakti Prima Dasa

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa Tugas Akhir dengan judul:

APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 22 Agustus 2011

Swara Adi Cakti Prima Dasa

3310801114

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801116

Nama : Fitra Fajrianto

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa Tugas Akhir dengan judul:

APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 22 Agustus 2011

Fitra Fajrianto

3310801116

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkah dan rahmat-NYA sehingga dapat menyelesaikan penyusunan Laporan Tugas Akhir ini dengan baik yang tidak terlepas dari bantuan dosen dan rekan-rekan mahasiswa Politeknik Negeri Batam.

Penulisan Laporan Tugas Akhir ini dimaksudkan untuk melengkapi persyaratan kelulusan tingkat Diploma III Program Studi Teknik Informatika Politeknik Negeri Batam. Untuk memenuhi persyaratan tersebut maka dibuatlah suatu aplikasi yang berjudul “APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*”.

OpenCV adalah singkatan dari *Open Computer Vision*, yaitu *library-library open source* yang di khususkan untuk melakukan *image prosessing*. Dengan menggunakan *OpenCV* memungkinkan sebuah *webcam* untuk mendeteksi sebuah objek (wajah), sehingga memungkinkan komputer untuk mengenali objek tersebut.

Penulisan Laporan Tugas Akhir ini dapat disusun dan diselesaikan dengan baik tidak terlepas dari bantuan dan dukungan dari semua pihak yang ikut dalam membimbing penulis untuk menyelesaikan Laporan Tugas Akhir ini. Pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Allah SWT, atas anugrah yang telah diberikan kepada penulis.
2. Nabi Muhammad SAW, sebagai inspirator hidup bagi penulis.
3. Ayahanda, Ibunda, Kakanda-kakandaku, dan semua keluarga atas doa, dukungan moril maupun materil, dan nasehat yang sangat berguna.
4. Bapak DR. Ir. Priyono Eko Sanyoto, M.Sc. Selaku direktur Politeknik Negeri Batam.

5. Bapak Uuf Brajawidagda, MT. Sebagai kepala Program Studi Teknik Informatika dan koordinator Tugas Akhir.
6. Bapak Riwinoto, M.Kom. Sebagai pembimbing dan pemberi semangat selama pembuatan Tugas Akhir ini.
7. Bapak/ Ibu dosen program studi Teknik Informatika Politeknik Negeri Batam dan seluruh pihak yang ikut serta membantu dan memberi semangat, sehingga matakuliah Tugas Akhir ini dapat diselesaikan.
8. Teman-teman Teknik Informatika angkatan 2008 yang saling mengingatkan, saling memberikan dukungan dan semangat satu sama yang lain.

Karena itu kami mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak semoga amal ibadah diterima di oleh Allah SWT, atas bantuan moril maupun spiritual demi terselesaikannya laporan ini. Penulis mohon maaf yang sebesar-besarnya kepada seluruh pihak atas kesalahan yang disengaja maupun yang tidak disengaja, penulis juga sadar masih banyak kekurangan yang terdapat pada penulisan Laporan Tugas Akhir ini, oleh karena itu saran dan kritik yang membangun penulis harapkan dari semua kalangan.

Batam, 22 Agustus 2011

Penulis

ABSTRAK

APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*

Tugas Akhir "Aplikasi Pengenalan Wajah Berbasis *OpenCV*" ini merupakan salah satu pengembangan fitur untuk melakukan pengenalan wajah terhadap suatu objek (wajah). Aplikasi Pengenalan Wajah ini dibuat untuk dapat mengatasi sulitnya mengenali seseorang, apabila orang tersebut memiliki nama yang sama dengan orang lain dan dapat mengakibatkan kesalahan dalam mengenali orang.

Dalam pembuatan Aplikasi Pengenalan Wajah Berbasis *OpenCV* ini akan membutuhkan *software* yang dapat melakukan *image processing*, *software* yang perlukan adalah *OpenCV*. Metode yang akan digunakan dalam pengolahan data wajah adalah *Face Matching*. Metode *Face Matching* berfungsi untuk memberikan keputusan dengan melihat tingkat kesamaan (*similarity*) pada data wajah yang lebih besar dari pada *thresholdnya*, kemungkinan sistem akan mengalami kesalahan dalam mengenali wajah seseorang. Adapun aplikasi pendukung yang digunakan dalam pembuatan aplikasi ini adalah *Microsoft Visual Studio 2010* dan bahasa pemrograman C# (C-Sharp) sebagai editor dalam pembuatan aplikasi.

Kata Kunci: *image processing*, *OpenCV*, *Face Matching*, *Microsoft Visual Studio 2010*, C# (C-Sharp).

ABSTRACT

APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*

End Task "Application-Based Face Recognition OpenCV" This is one of the development features to perform face recognition to an object (face). Face Recognition Application are made to be able to overcome the difficulty of recognizing a person, if that person the have same name with others and will cause errors in recognizing people.

In the manufacture of Application-Based Face Recognition OpenCV This will require software that can perform image processing, software used is OpenCV. Methods to be used in data processing is a Face Matching faces. Face Matching method serves to provide a decision by looking at the degree of similarity (similarity) in the face of data that is greater than the thresholdnya, the system likely will experience an error in recognizing a person's face. Supporting applications that are used in making this application are Microsoft Visual Studio 2010 and programming languages C# (C-Sharp) as editor in making the application.

Keywords: image processing, OpenCV, Face Matching, Microsoft Visual Studio 2010, C# (C-Sharp).

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERNYATAAN	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR.....	iiiv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
Bab I Pendahuluan.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	2
I.3 Batasan Masalah	2
I.4 Tujuan	3
I.5 Sistematika Penulisan	3
Bab II Landasan Teori	5
II.1 Pengenalan Objek Sistem Pengenalan Wajah	5
II.1.1 Pemrosesan Awal pada Sistem Pengenalan Wajah	6
II.1.2 Proses Konversi Citra RGB ke Grayscale Image	7
II.1.2.1 Citra RGB.....	9
II.1.2.2 Citra Grayscale	11
II.1.3 Proses Histogram Equalization.....	11
II.1.4 Proses Face Matching Menggunakan Thresholding.....	14
II.1.5 Proses Pengenalan Wajah	16
II.2 FaceSDK.....	17
II.3 Hubungan Antar Proses	19
Bab III Analisis	20

III.1	Deskripsi Umum Sistem	20
III.2	Fitur Utama Perangkat Lunak	20
III.2.1	Kebutuhan Fungsional	21
III.3	Use Case	21
III.3.1	Diagram Use Case	21
III.3.2	Skenario Use Case	21
III.3.2.1	Skenario Use Case Start	21
III.3.2.2	Skenario Use Case Remember All	22
III.3.2.3	Skenario Use Case Input Name	22
III.3.2.4	Skenario Use Case Pengujian	22
III.4	Analisis Kelas	23
Bab IV	Perancangan Sistem dan Implementasi	24
IV.1	Diagram Interaksi	24
IV.1.1	Diagram Interaksi <i>Use Case</i> Remember All	24
IV.1.2	Diagram Interaksi Use Case Pengujian	25
IV.2	Diagram Kelas	25
IV.3	Rancangan Kelas Rinci	26
IV.3.1	Kelas Pengenalan Wajah	26
IV.3.2	Kelas Start	26
IV.3.3	Kelas Remember All	26
IV.3.4	Kelas Pengujian	26
IV.3.5	Kelas Image	26
IV.3.6	Kelas Name	26
IV.3.7	Kelas Image Uji	27
IV.4	Algoritma	27
IV.4.1	Algoritma Pada Kelas Remember All	27
IV.4.2	Algoritma Pada Kelas Pengujian	27
IV.5	Perancangan Antarmuka	28
IV.5.1	Rancangan Tampilan Menu Utama	29
IV.5.2	Rancangan Tampilan Aplikasi Pengenalan Wajah	29
IV.5.3	Rancangan Tampilan Start	30

IV.5.4	Rancangan Tampilan Remember All.....	30
IV.5.5	Rancangan Tampilan Pengujian	31
Bab IV	Implementasi dan Pengujian.....	32
V.1	Implementasi dan Pengujian Aplikasi	32
V.1.1	Implementasi Kelas	32
V.1.2	Implementasi Antarmuka	32
V.2	Pengujian.....	33
V.2.1	Skenario Pengujian.....	33
V.2.2	Hasil Pengujian	33
Bab IV	Kesimpulan dan Saran	37
VI.1	Kesimpulan	37
VI.2	Saran.....	37
DAFTAR PUSTAKA	38

DAFTAR GAMBAR

Gambar II.1	Proses Tahap Pengambilan dan Pengenalan Wajah	6
Gambar II.2	Proses Tahap Pemrosesan Awal	7
Gambar II.3	Konversi RGB ke <i>Grayscale</i>	8
Gambar II.4	Representasi format citra RGB	10
Gambar II.5	Pengaturan citra RBG	10
Gambar II.6	Citra <i>Grayscale</i> dan Citra Hitam Putih	11
Gambar II.7	Contoh Histogram	12
Gambar II.8	Histogram Kumulatif	13
Gambar II.9	Hasil Histogram <i>Equalization</i>	14
Gambar II.10	Proses <i>Face Matching</i>	15
Gambar II.11	Proses Pengenalan Wajah	16
Gambar III.1	Deskripsi Umum Sistem	20
Gambar III.2	Diagram <i>Use Case</i>	21
Gambar III.3	Analisis Kelas	23
Gambar IV.1	Diagram Interaksi <i>Use Case</i> Remember All	24
Gambar IV.2	Diagram Interaksi <i>Use Case</i> Pengujian	25
Gambar IV.3	Diagram Kelas	25
Gambar IV.4	Rancangan Tampilan Menu Utama	29
Gambar IV.5	Rancangan Tampilan Aplikasi Pengenalan Wajah	29
Gambar IV.6	Rancangan Tampilan Start	30
Gambar IV.7	Rancangan Tampilan Remember All	30
Gambar IV.8	Rancangan Tampilan Pengujian	31
Gambar V.1	Rancangan Tampilan Pengujian 1	33
Gambar V.2	Rancangan Tampilan Pengujian 2	34
Gambar V.3	Rancangan Tampilan Pengujian 3	35

DAFTAR TABEL

Tabel II.1	Proses Perhitungan Distribusi Kumulatif pada Histogram	12
Tabel II.2	Proses Teknik Perataan Histogram	13
Tabel III.1	Spesifikasi Analisis Kelas	23
Tabel V.2	Implementasi Kelas	32
Tabel V.2	Implementasi Antarmuka	32

ABSTRAK

APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*

Tugas Akhir "Aplikasi Pengenalan Wajah Berbasis *OpenCV*" ini merupakan salah satu pengembangan fitur untuk melakukan pengenalan wajah terhadap suatu objek (wajah). Aplikasi Pengenalan Wajah ini dibuat untuk dapat mengatasi sulitnya mengenali seseorang, apabila orang tersebut memiliki nama yang sama dengan orang lain dan dapat mengakibatkan kesalahan dalam mengenali orang.

Dalam pembuatan Aplikasi Pengenalan Wajah Berbasis *OpenCV* ini akan membutuhkan *software* yang dapat melakukan *image processing*, *software* yang perlukan adalah *OpenCV*. Metode yang akan digunakan dalam pengolahan data wajah adalah *Face Matching*. Metode *Face Matching* berfungsi untuk memberikan keputusan dengan melihat tingkat kesamaan (*similarity*) pada data wajah yang lebih besar dari pada *thresholdnya*, kemungkinan sistem akan mengalami kesalahan dalam mengenali wajah seseorang. Adapun aplikasi pendukung yang digunakan dalam pembuatan aplikasi ini adalah *Microsoft Visual Studio 2010* dan bahasa pemrograman C# (C-Sharp) sebagai editor dalam pembuatan aplikasi.

Kata Kunci: *image processing*, *OpenCV*, *Face Matching*, *Microsoft Visual Studio 2010*, C# (C-Sharp).

ABSTRACT

APLIKASI PENGENALAN WAJAH BERBASIS *OPENCV*

End Task "Application-Based Face Recognition OpenCV" This is one of the development features to perform face recognition to an object (face). Face Recognition Application are made to be able to overcome the difficulty of recognizing a person, if that person the have same name with others and will cause errors in recognizing people.

In the manufacture of Application-Based Face Recognition OpenCV This will require software that can perform image processing, software used is OpenCV. Methods to be used in data processing is a Face Matching faces. Face Matching method serves to provide a decision by looking at the degree of similarity (similarity) in the face of data that is greater than the thresholdnya, the system likely will experience an error in recognizing a person's face. Supporting applications that are used in making this application are Microsoft Visual Studio 2010 and programming languages C# (C-Sharp) as editor in making the application.

Keywords: image processing, OpenCV, Face Matching, Microsoft Visual Studio 2010, C# (C-Sharp).

Bab I Pendahuluan

Pada bab pendahuluan ini akan dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan serta sistematika penulisan.

I.1 Latar Belakang

Perkembangan teknologi informasi saat ini semakin maju, teknologi yang dimilikipun semakin beragam dan diminati untuk digunakan di hampir semua bidang. Salah satu bidang yang sekarang ini mulai dikembangkan adalah biometrik. Biometrik adalah sistem pengenalan yang bekerja dengan mengambil data-data biometrika dari individu tertentu, data yang diproses pada biometrik adalah ciri-ciri khusus fisik yang dimiliki seseorang, salah satu bentuk data biometrik adalah pengenalan wajah (*face recognition*).

Proses pengenalan wajah (*face recognition*), diawali dengan proses pendeteksian wajah, proses pendeteksian wajah memiliki peran yang sangat penting dalam sistem pengenalan wajah, hal itu dikarenakan proses pengenalan akan lebih akurat setelah wajah dalam suatu gambar telah terdeteksi. Pendeteksian wajah melalui *webcam* dapat digunakan juga untuk mengumpulkan wajah yang lebih bervariasi. Jika pendeteksian dilakukan secara terus-menerus maka variasi wajah yang tersimpan pun akan semakin bervariasi sehingga akan membantu dalam proses pengenalan wajah. Sistem pengenalan bertujuan memecahkan identitas seseorang. Terdapat dua tipe sistem pengenalan, yaitu sistem verifikasi dan identifikasi. Sistem verifikasi bertujuan untuk menerima atau menolak identitas yang diklaim oleh seseorang, sedangkan sistem identifikasi bertujuan untuk memecahkan identitas seseorang.

Salah satu teknologi yang dapat digunakan untuk membuat aplikasi tersebut, yaitu software yang dinamakan *OpenCV*. *OpenCV* adalah singkatan dari *Open Computer Vision*, yaitu *library-library open source* yang di khususkan untuk melakukan *image processing*. Dengan menggunakan *OpenCV* memungkinkan

webcam untuk mendeteksi sebuah objek (wajah), sehingga memungkinkan komputer untuk mengenali objek tersebut. Editor untuk melakukan pembuatan aplikasi ini yaitu dengan menggunakan bahasa pemrograman C# pada *Microsoft Visual Studio 2010 Express Edition*.

Sebelumnya telah ada penelitian oleh M.Taufik Hidayat (2005-2006) dalam pembuatan Tugas Akhir mengenai *face recognition* yang berjudul perbandingan citra (studi kasus wajah manusia), yang dapat mengenali identitas seseorang dengan melihat jarak antar titik yang ada disekitar mata. Untuk mengenali seseorang dengan aplikasi tersebut masih bersifat manual sehingga memakan waktu yang cukup lama dan masih rentan kesalahan dikarenakan pengenalan hanya difokuskan pada jarak antar titik disekitar mata saja.

Hal itulah yang melatarbelakangi dalam pembuatan Tugas Akhir ini, untuk dapat melakukan pengembangan terhadap aplikasi yang telah dibuat sebelumnya. Pembuatan Tugas Akhir ini akan diberi judul “Aplikasi Sistem Pengenalan Wajah Pada Manusia” dengan memfokuskan terhadap beberapa titik diseluruh wajah sehingga pengenalan dapat dilakukan dengan lebih akurat.

I.2 Rumusan Masalah

Adapun rumusan masalah dari aplikasi ini adalah :

1. Bagaimana melakukan pendeteksian wajah.
2. Bagaimana mengenali wajah yang telah terdeteksi.

I.3 Batasan Masalah

Pada Aplikasi Sistem Pengenalan Wajah ini memiliki batasan masalah 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. Pendeteksian dan pengenalan wajah tidak bisa dilakukan ditempat yang gelap.
3. Pendeteksian dan pengenalan wajah tidak dapat dilakukan untuk dua orang atau lebih secara bersamaan.

I.4 Tujuan

Tujuan dari aplikasi ini adalah :

1. Melakukan pendeteksian wajah melalui *webcam*.
2. Mengidentifikasi atau mengenali wajah yang terdeteksi oleh *webcam*.

I.5 Sistematika Penulisan

Sistematika penulisan laporan ini terdiri atas:

Bab I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

Bab II Landasan Teori

Berisi tentang teori-teori yang berkaitan dengan Tugas Akhir dan juga penjelasan-penjelasan untuk membantu dalam pembuatan Tugas Akhir.

Bab III Analisis

Berisi tentang deskripsi umum sistem, *usecase*, serta analisis kelas.

Bab IV Perancangan Sistem

Berisi tentang diagram interaksi, diagram kelas, rancangan kelas rinci dan rancangan antarmuka.

Bab V Implementasi dan Pengujian

Setelah dilakukan tahap perancangan maka tahap selanjutnya adalah implementasi dan pengujian pada perangkat lunak. Implementasi akan menghasilkan aplikasi yang dapat dijalankan di lingkungan operasional. Untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai deskripsi perencanaan maka perlu dilakukan pengujian.

Bab VI Kesimpulan dan Saran

Berisi tentang kesimpulan dari studi kasus Aplikasi Sistem Pengenalan Wajah Berbasis *OpenCV*, dan saran untuk pengembangan aplikasi ini selanjutnya.

Bab II Landasan Teori

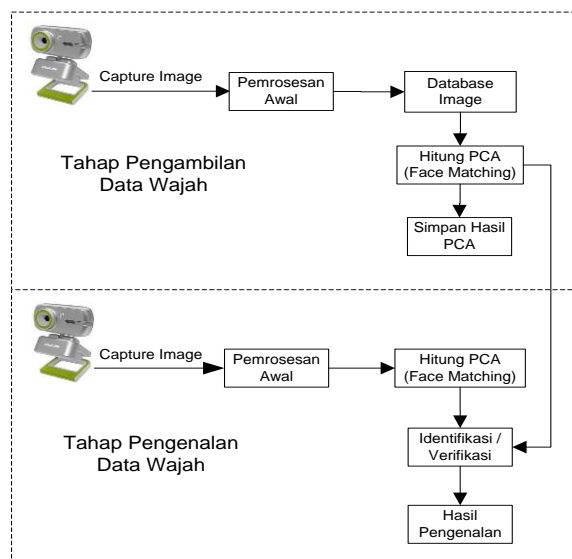
Pada bab ini berisi tentang teori-teori yang berkaitan dengan Tugas Akhir dan juga penjelasan-penjelasan yang digunakan untuk membantu dalam pembuatan Tugas Akhir.

II.1 Pengenalan Objek Sistem Pengenalan Wajah

Pada bab ini ada beberapa keterangan dasar yang akan dijelaskan tentang sistem pengenalan wajah menggunakan *webcam*. Pada sistem proses pengenalan ini dibagi menjadi 2 tahap, yaitu tahap pengambilan data wajah dan tahap pengenalan wajah. Pada tahap pengambilan data wajah, proses pertama yang dilakukan yaitu melakukan pengambilan gambar secara langsung melalui *webcam*. Setelah melakukan pengambilan gambar pada wajah, selanjutnya dilakukan pemrosesan awal yang meliputi, konversi format RGB ke dalam format *grayscale* yang dimana hasilnya adalah hitam putih. Proses selanjutnya yaitu normalisasi ukuran citra (*resize*), proses ini digunakan untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses. Kemudian akan dilakukan proses *histogram equalization* yang digunakan untuk memperbaiki kualitas citra gambar agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya. Selanjutnya akan dilakukan proses PCA (*Principal Component Analysis*) untuk mengambil bagian terpenting dari wajah dengan metode *face matching* sehingga aplikasi dapat melakukan sistem verifikasi dan identifikasi terhadap data wajah yang diuji.

Sedangkan untuk tahap pengenalan wajah, proses pertama yang akan dilakukan yaitu melakukan pengambilan gambar untuk menampilkan objek yang ditangkap oleh *webcam*. Kemudian system akan mendeteksi wajah dari objek yang ada melalui *webcam*. Selanjutnya dilakukan pemrosesan awal yang meliputi, normalisasi ukuran citra, RGB ke *grayscale*, *histogram equalization* untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa

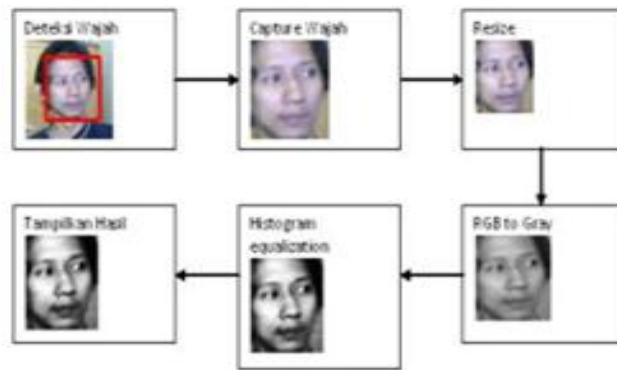
menghilangkan informasi utamanya, *resize* untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input. Kemudian dilakukan proses PCA untuk mengutip bagian terpenting dengan metode *face matching* sehingga didapatkan *facial feature detection* dan *eye centers detection* dari gambar tersebut. Pada proses akhir akan dilakukan pengenalan wajah dengan menghitung jarak antara fitur wajah yang ada didalam sistem dan fitur wajah dari data uji. Jarak antar fitur yang didapat di cari yang paling mendekati atau sama, kemudian dilakukan proses verifikasi dan identifikasi.



Gambar II.1 Proses Tahap Pengambilan dan Pengenalan Wajah

II.1.1 Pemrosesan Awal pada Sistem Pengenalan Wajah

Pada saat proses deteksi wajah dilakukan maka langsung dilakukan pemrosesan awal saat citra wajah dalam frame terdeteksi. Hal ini dilakukan untuk membuat software yang *realtime* dan lebih efisien. Proses pertama yang dilakukan adalah melakukan pendeteksian wajah, selanjutnya melakukan pengambilan gambar. Kemudian proses selanjutnya adalah mengubah format RGB ke dalam format *Grayscale*. Setelah bentuk format diubah, kemudian akan dilakukan proses *histogram equalization*. Pada proses akhir dari pemrosesan awal ini adalah menampilkan hasil.



Gambar II.2 Proses Tahap Pemrosesan Awal

Proses pendeteksian terjadi saat adanya citra wajah dalam gambar dengan detektor wajah menggunakan *OpenCV*.

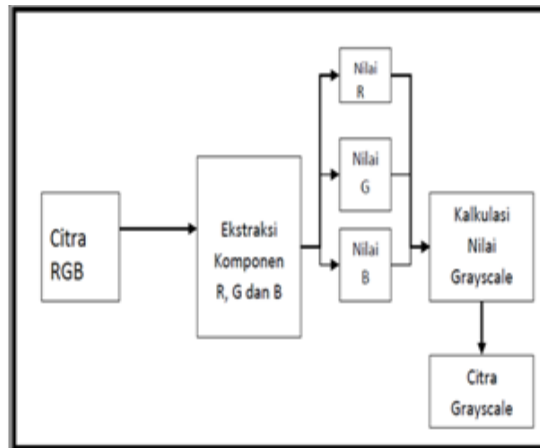
- a. Pengambilan gambar setelah wajah terdeteksi.
- b. Citra wajah yang diperoleh dari berbagai sumber memiliki ukuran yang beragam, karena itu harus diseragamkan sehingga memiliki ukuran 50 x50 piksel.
- c. Citra wajah diubah kedalam grayscale sebelum dilakukan *histogram equalization*.
- d. Pada tahap ini citra yang telah diubah dalam *grayscale* diperbaiki distribusi nilai derajat keabuan dengan cara dibuat rata.

II.1.2 Proses Konversi Citra RGB ke Grayscale Image

Konversi citra dari format RGB ke dalam format *grayscale* sangat diperlukan untuk keperluan pemrosesan citra, diantaranya untuk memperkecil ukuran file dan untuk memenuhi syarat beberapa algoritma identifikasi yang memerlukan citra matriks tunggal¹.

¹Teknik mengubah format citra .bmp yang bersifat RGB ke format Grayscale.

<http://p3m.amikom.ac.id/p3m/51%20%20KONVERSI%20FORMAT%20CITRA%20RGB%20KE%20FORMAT%20GRAYSCALE.pdf>



Gambar IL3 Konversi RGB ke Grayscale

Proses pertama adalah mengambil nilai R, G dan B dari suatu citra bertipe RGB. Pada tipe bmp citra direpresentasikan dalam 24 bit, sehingga diperlukan proses untuk mengambil masing - masing 3 kelompok 8 bit dari 24 bit tadi. Sebagai contoh suatu *pixel* memiliki nilai RGB 24 bit sebagai berikut :

111100001111000011111111, untuk mendapatkan masing-masing nilai R, G dan B dilakukan operasi-operasi sebagai berikut.

Untuk mendapatkan nilai R dilakukan operasi modulo dengan bilangan 256 sebagai berikut :

$$\begin{aligned} \text{Nilai R} &= 111100001111000011111111 \bmod 10000000 \\ &= 11111111 \end{aligned}$$

Sedangkan untuk nilai G, dapat dicari dengan cara sebagai berikut:

$$\begin{aligned} \text{Nilai G} &= (111100001111000011111111 \text{ dan } 1111111100000000) / 100000000 \\ &= 11110000 \end{aligned}$$

Untuk Nilai B, dapat dicari dengan menggunakan rumus

$$\begin{aligned} \text{Nilai B} &= (111100001111000011111111 \text{ dan } \\ &111111110000000000000000) / 10000000000000000 \end{aligned}$$

$$= 11110000$$

Sehingga dari nilai pixel 1111000011110000111111112 atau 15790335 diperoleh nilai :

$$R = 11111111 = 255$$

$$G = 11110000 = 240$$

$$B = 11110000 = 240$$

Sehingga diperoleh triplet RGB= (255,240,240).

Setelah nilai *triplet* RGB diperoleh, maka kita bisa mendapatkan nilai *grayscale* dari *pixel* tersebut. Ide dasarnya sebenarnya adalah membuat band tunggal dari 3 band RGB tadi dengan rumus tertentu. Pada penelitian ini digunakan rumus :

$$\text{red} = (\text{red} * 5) \setminus 10$$

$$\text{green} = (\text{green} * 8) \setminus 10$$

$$\text{blue} = (\text{blue} * 3) \setminus 10$$

$$\text{gray} = ((\text{red} + \text{green} + \text{blue}) * 10) \setminus 16$$

Dengan mengaplikasikan prosedur tadi pada semua *pixel* akan kita dapatkan citra dengan format *grayscale*².

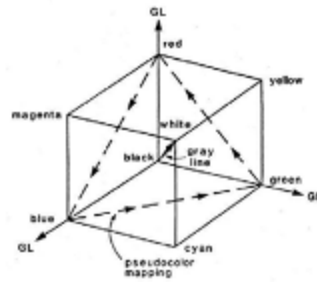
II.1.2.1 Citra RGB

Suatu citra biasanya mengacu ke citra RGB. Sebenarnya bagaimana citra disimpan dan dimanipulasi dalam komputer diturunkan dari teknologi televisi, yang pertama kali mengaplikasikannya untuk tampilan grafis komputer. Jika dilihat dengan kaca pembesar, tampilan monitor komputer akan terdiri dari

²Proses pengubahan citra RGB ke dalam citra Grayscale.

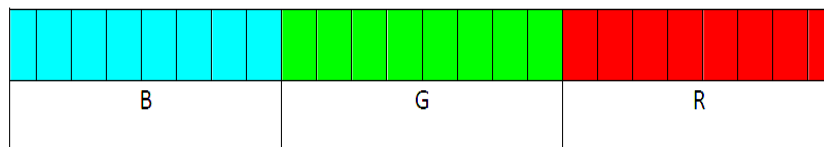
<http://p3m.amikom.ac.id/p3m/51%20%20KONVERSI%20FORMAT%20CITRA%20RGB%20KE%20FORMAT%20GRAYSCALE.pdf>

sejumlah triplet titik warna merah (*red*), hijau (*green*) dan biru (*blue*). Tergantung pada pabrik monitornya untuk menentukan apakah titik tersebut merupakan titik bulat atau kotak kecil, akan tetapi selalu terdiri dari 3 *triplet* yaitu *red*, *green* dan *blue*.



Gambar IL4 Representasi format citra RGB

Citra dalam komputer tidak lebih dari sekumpulan sejumlah triplet dimana setiap *triplet* terdiri atas variasi tingkat keterangan (*brightness*) dari elemen *red*, *green* dan *blue*. Representasinya dalam citra, *triplet* akan terdiri dari 3 angka yang mengatur intensitas dari *Red* (R), *Green* (G) dan *Blue* (Blue) dari suatu *triplet*. Setiap *triplet* akan merepresentasikan 1 pixel (*picture element*). Suatu *triplet* dengan nilai 67, 228 dan 180 berarti akan mengeset nilai R ke nilai 67, G ke nilai 228 dan B ke nilai 180. Angka-angka RGB ini yang seringkali disebut dengan *color values*. Pada format .bmp citra setiap pixel pada citra direpresentasikan dengan 24 bit, 8 bit untuk R, 8 bit untuk G dan 8 bit untuk B. Bentuk pengaturan citra RGB seperti pada gambar IL.5³.



Gambar IL5 Pengaturan citra RBG

³Cara memanipulasikan citra RGB di dalam komputer.

<http://p3m.amikom.ac.id/p3m/51%20%20KONVERSI%20FORMAT%20CITRA%20RGB%20K%20FORMAT%20GRAYSCALE.pdf>

II.1.2.2 Citra Grayscale

Dalam komputasi, suatu citra digital *grayscale* atau *greyscale* adalah suatu citra dimana nilai dari setiap pixel merupakan sample tunggal. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih". Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik *single band*⁴.

Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak.



Gambar II.6 Citra *Grayscale* dan Citra Hitam Putih

II.1.3 Proses Histogram Equalization

Histogram Equalization adalah suatu proses perataan *histogram*, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Untuk dapat melakukan

⁴Perbedaan warna dan variasi warna citra yang terdapat pada citra RGB.

<http://p3m.amikom.ac.id/p3m/51%20%20KONVERSI%20FORMAT%20CITRA%20RGB%20K%20FORMAT%20GRAYSCALE.pdf>

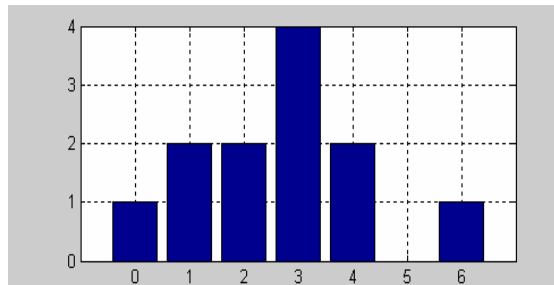
histogram equalization ini diperlukan suatu fungsi distribusi kumulatif yang merupakan kumulatif dari *histogram*⁵.

Contoh proses *histogram equalization* sebagai berikut :

Misalkan diketahui data sebagai berikut:

2 4 3 1 3 6 4 3 1 0 3 2

Maka *histogram* dari data di atas adalah:



Gambar IL.7 Contoh *Histogram*

Proses perhitungan distribusi kumulatif dapat dijelaskan dengan tabel berikut:

Tabel IL.1 Proses Perhitungan Distribusi Kumulatif pada *Histogram*

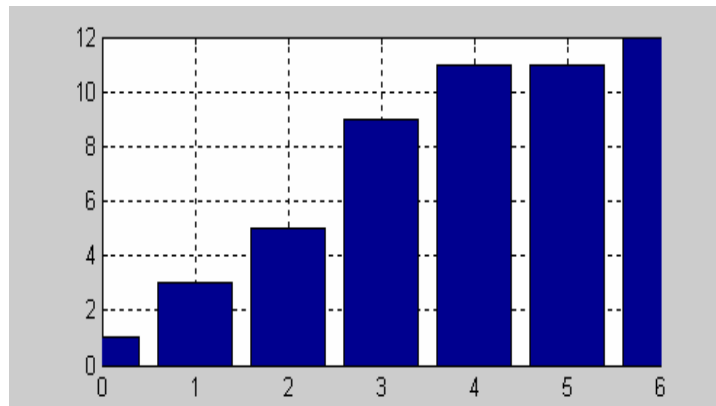
Nilai	Histogram	Distribusi kumulatif
0	1	1
1	2	1+2=3
2	2	3+2=5
3	4	5+4=9
4	2	9+2=11
5	0	11+0=11
6	1	11+1=12

⁵Pengertian *histogram equalization*.

[http://lecturer.eepisits.edu/~nana/index_files/materi/Prak_Citra/Praktikum% 206% 20Histogram.pdf](http://lecturer.eepisits.edu/~nana/index_files/materi/Prak_Citra/Praktikum%206%20Histogram.pdf)

f

Dan diperoleh *histogram* kumulatif sebagai berikut:



Gambar II.8 Histogram Kumulatif

Histogram equalization (perataan *histogram*) adalah suatu proses dimana *histogram* diratakan berdasarkan suatu fungsi linier (garis lurus) seperti terlihat pada gambar II.8. Teknik perataan *histogram* adalah sebagai berikut :

Tabel II.2 Proses Teknik Perataan *Histogram*

Nilai Asal	Histogram Kumulatif	Nilai hasil
0	1	$1/2 \rightarrow 0$
1	3	$3/2 \rightarrow 1$
2	5	$5/2 \rightarrow 2$
3	9	$9/2 \rightarrow 4$
4	11	$11/2 \rightarrow 5$
5	11	$11/2 \rightarrow 5$
6	12	$12/2 \rightarrow 6$

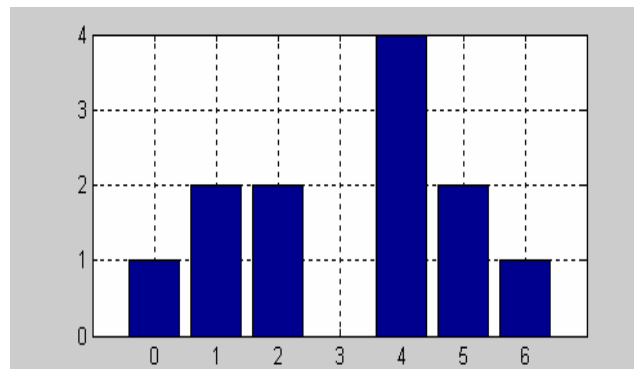
Untuk mendapatkan hasil *histogram equalization* digunakan rumus sebagai berikut :

$$w = \frac{c_w \cdot th}{n_x n_y}$$

- Keterangan :
- w = nilai keabuan hasil *histogram equalization*
 - c_w = *histogram* kumulatif dari w
 - th = threshold derajat keabuan (256)
 - n_x dan n_y = ukuran gambar

Setelah melakukan perhitungan untuk mencari hasil *histogram equalization*, di dapat hasil sebagai berikut :

2 5 4 1 4 6 5 4 1 0 4 2



Gambar II.9 Hasil *Histogram Equalization*

II.1.4 Proses Face Matching Menggunakan Thresholding

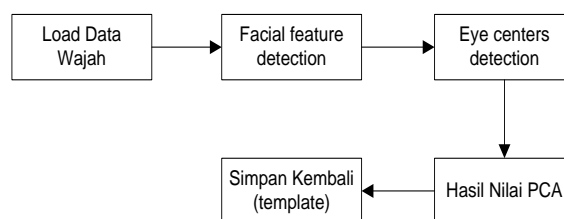
Prinsip dasar dari proses pengenalan wajah ini adalah dengan melakukan pencocokan antara data wajah yang disimpan dengan data wajah yang akan diuji. Fungsi *face matching* ini mengembalikan tingkat kesamaan wajah. Sistem mempertimbangkan tingkat kesamaan yang paling mirip adalah sama, sehingga

sistem menganggap bahwa data wajah disimpan dan data wajah yang diuji adalah milik satu orang yang sama.

Metode *face matching* ini akan memberikan dua hasil yaitu FAR(*False Acceptance Rate*) dan FRR(*False Rejection Rate*), untuk ukuran data wajah yang disimpan adalah sebesar 16 Kb, proses pencocokan wajah antara data yang telah disimpan dan data uji pada metode ini dapat mencocokkan sebanyak 30.000 data wajah per detiknya, sehingga proses pencocokan hampir tidak memakan waktu. Proses *face matching* ini mengacu pada hasil dari proses *facial feature detection* dan *eye centers detection*⁶.

Proses *facial feature detection* ini berfungsi untuk melakukan pendeteksian wajah yang berguna untuk proses pengenalan wajah nantinya. Pertama, proses ini mendeteksi sebuah wajah frontal pada gambar, dan kemudian mendeteksi fitur wajah. Proses ini dapat mendeteksi kurang lebih 40 titik fitur wajah yang ada pada wajah (mata, alis, mulut, hidung, kontur wajah), untuk melakukan pendeteksian ini memakan waktu sebanyak 0.59 detik (tidak termasuk tahap pendeteksian wajah)⁷.

Untuk proses *eye centers detection* bekerja jauh lebih cepat dari pada proses *facial feature detection*, untuk proses ini akan memakan waktu sekitar 0.067 detik (tidak termasuk tahap pendeteksian wajah)⁸.



Gambar II.10 Proses Face Matching

⁶ Penggunaan metode *face matching* pada *FaceSDK*.

<http://luxand.com/facesdk/documentation/specifications.php>

⁷ Proses *facial feature detection* yang terdapat di dalam metode *face matching*.

<http://luxand.com/facesdk/documentation/specifications.php>

⁸ Proses *eye centers detection* yang terdapat di dalam metode *face matching*.

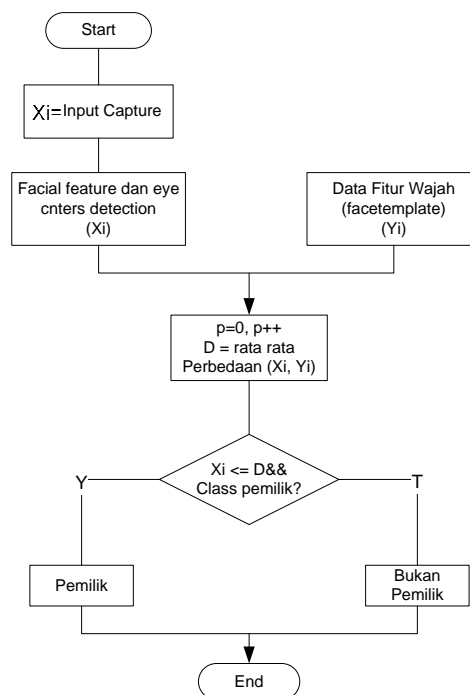
<http://luxand.com/facesdk/documentation/specifications.php>

Proses PCA dilakukan dengan :

1. Mengambil data dari hasil training.
2. Melakukan proses *facial feature detection* untuk menemukan nilai PCA.
3. Melakukan proses *eye centers detection* untuk menemukan nilai PCA.
4. Mendapatkan nilai PCA.
5. Menyimpan semua informasi hasil pelatihan didalam *template*.

II.1.5 Proses Pengenalan Wajah

Dalam proses pengenalan wajah ini menggunakan metode *face matching* yang mengacu pada proses *facial feature detection* dan *eye centers detection*. *Face matching* disini akan berguna dalam pemberian keputusan antara pemilik dan bukan pemilik dari data wajah yang tersimpan didalam sistem berdasarkan dari data wajah yang diuji. Diagram alur keseluruhan dalam proses pengenalan wajah dapat dilihat pada gambar II.11.



Gambar II.11 Proses Pengenalan Wajah

Pada gambar di atas adalah proses dimana pengenalan wajah dilakukan. Proses pengenalannya adalah sebagai berikut :

Pertama adalah ambil sebuah citra data wajah, kemudian tentukan jarak antar titik yang ditemukan pada *facial feature detection* dan *eye centers detection*.. Selanjutnya hitung tingkat kemiripan (*similarity*) menggunakan *face matching* antara fitur input (X_i) dan fitur hasil pelatihan (Y_i), jika jarak dari fitur input (X_i) lebih kecil dari pada rata-rata perbedaan dan nilai tersebut masuk kedalam *class* pemilik maka diidentifikasi sebagai pemilik dan jika jarak dari fitur input (X_i) lebih besar dari pada rata-rata perbedaan dan berada diluar *class* pemilik maka diidentifikasi sebagai bukan pemilik.

Sistem pengenalan wajah dapat dilakukan dengan menggunakan metode *face matching* yang berfungsi untuk memberikan keputusan dengan melihat tingkat kesamaan(*similarity*) pada data wajah yang lebih besar dari pada *thresholdnya*, kemungkinan sistem salah dalam mengenali wajah seseorang adalah $1-X$. Sebagai contoh, jika keputusan untuk menyatakan bahwa wajah tersebut dikenali didasarkan pada *code*:

```
If (Similarity > 0.99)
    { match = true ; }
```

Berdasarkan *code* diatas, maka kemungkinan sistem salah dalam mengenali atau mencocokkan wajah adalah 0.01 atau sekitar 1%.

II.2 FaceSDK

FaceSDK adalah cross-platform untuk deteksi dan pengenalan wajah agar *library-library image processing* dapat digunakan pada aplikasi pelanggan. *FaceSDK* ini sendiri merupakan pembungkus dari *OpenCV library* agar dapat digunakan di bahasa pemrograman C#. *FaceSDK* menawarkan API (*Application Programming interface*) untuk mendeteksi wajah dan melakukan *face matching* setelah wajah terdeteksi oleh *webcam*. *FaceSDK* menyediakan koordinat 40 titik fitur pada

wajah yang akan diproses lebih lanjut agar dapat diukur tingkat kemiripannya dengan data uji nantinya. Luxand *FaceSDK* tersedia untuk semua mulai dari 32-bit dan 64-bit dari windows dan linux, dan 64-bit MacOS X⁹.

Persyaratan agar *library FaceSDK* dapat digunakan:

- Windows 2000/XP/2003/Vista/2008/Seven
- Linux (RHEL 5 +, CentOS 5 + dan lainnya)
- Mac OS X 10.4 + x86_64

Persyaratan sistem minimum:

- Prosesor 1,6 GHz
- 256 MB RAM
- 150 MB ruang disk bebas

Fitur Persyaratan sistem:

- 2,4 GHz Intel multi-core
- 2 GB RAM
- DirectShow-kompatibel webcam

FaceSDK merupakan platform berbayar yang disediakan oleh Luxand, kita dapat membuat atau mendistribusikan salinan terbatas versi *trial* produk pada kode objek saja. Maka dalam pembuatan Aplikasi Pengenalan Wajah ini digunakanlah *platform FaceSDK* dari Luxand versi *trial* yang bertujuan untuk mengetahui dan mengevaluasi kinerja dari *FaceSDK* ini terlebih dahulu.

⁹ Pengertian *FaceSDK*.

<http://luxand.com/facesdk/documentation/overview.php>

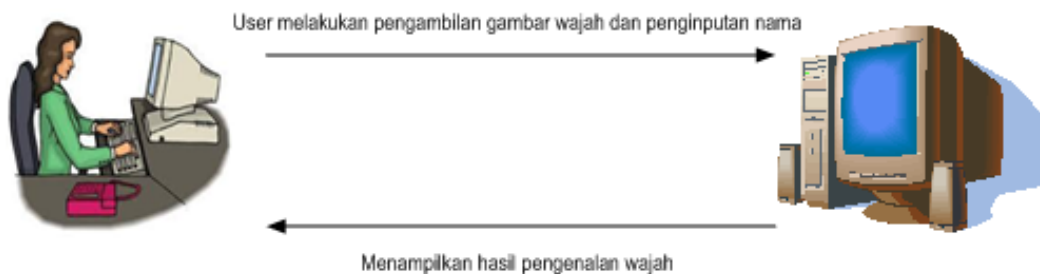
II.3 Hubungan Antar Proses

Hubungan antar proses untuk menyelesaikan Tugas Akhir ini yaitu proses pertama mengambil data wajah yang terdeteksi oleh webcam dengan format RGB, kemudian format RGB dikonversi ke dalam format *grayscale* dimana hasilnya berupa *image* hitam putih, proses selanjutnya yaitu *facial feature detection* dan *eye centers detection* yang berguna untuk mendeteksi 40 titik fitur yang ada pada data wajah (mata, alis, mulut, hidung, kontur wajah). Setelah semua dilakukan maka pada saat melakukan pengujian (pengenalan wajah), digunakanlah metode *face matching*. Metode ini berfungsi untuk mencocokkan atau membandingkan antara data uji dan data yang telah tersimpan dan memberikan hasil apakah data yang diuji tersebut dikenali atau tidak.

Bab III ANALISIS

Pada bab analisis ini akan dijelaskan tentang deskripsi umum sistem, fitur utama perangkat lunak, *use case* serta analisis kelas.

III.1 Deskripsi Umum Sistem



Gambar III. 1 Deskripsi Umum Sistem

Keterangan :

User melakukan pengambilan gambar wajah dan penginputan nama, kemudian data wajah akan disimpan ke dalam sistem yang memuat data wajah, dan data wajah disimpan sebanyak 10 data per *user*nya dan data nama akan disimpan kedalam sistem yang memuat data nama. Setelah itu data yang diambil akan dijadikan sebagai acuan untuk diolah antara saat pengambilan gambar dilakukan dengan data yang sudah ada didalam sistem. Kemudian apabila data tersebut terdapat kesamaan, maka pengenalan wajah berhasil dilakukan.

III.2 Fitur Utama Perangkat Lunak

Aplikasi ini telah disesuaikan dengan kebutuhan umum permasalahan yang telah diangkat. Aplikasi ini dapat digunakan untuk melakukan pengenalan wajah.

III.2.1 Kebutuhan Fungsional

F-001 Sistem menerima masukkan data berupa gambar wajah dari *user*.

F-002 Sistem mampu untuk merubah data masukkan dari RGB menjadi data *grayscale*.

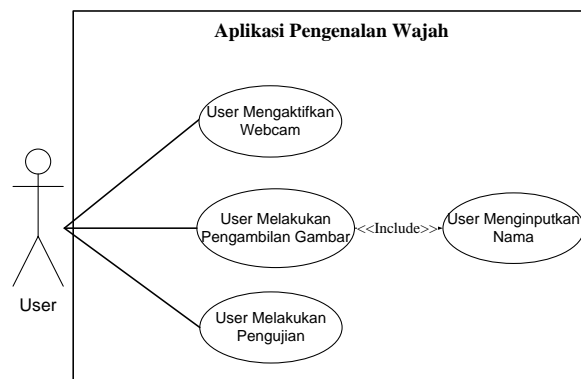
F-003 Sistem mampu mencari, mengolah dan menggolongkan data-data yang memiliki kemiripan.

F-003 Sistem mampu menampilkan informasi pengenalan wajah tersebut.

III.3 Use Case

III.3.1 Diagram Use Case

Diagram *use case* adalah diagram yang menggambarkan interaksi antara pengguna dengan sistem, maka dapat disimpulkan diagram *use case* dari Aplikasi Sistem Pengenalan Wajah berbasis *OpenCV* adalah sebagai berikut:



Gambar III.2 Diagram Use case

III.3.2 Skenario Use Case

III.3.2.1 Skenario Use Case Start

Kondisi awal : Webcam belum di aktifkan

Kondisi akhir : Webcam telah di aktifkan

Skenario : User dapat mengaktifkan webcam dengan mengklik tombol start.

III.3.2.2 Skenario Use Case Remember All

- Kondisi awal : *Webcam* telah di aktifkan, namun data wajah dan nama belum diambil atau disimpan
- Kondisi akhir : Data wajah dan nama telah di simpan
- Skenario : *User* dapat mengklik tombol *remember all* untuk mengambil gambar sebanyak 10 gambar per user, data tersebut akan disimpan kedalam sistem yang memuat data wajah.

III.3.2.3 Skenario Use Case Input Name

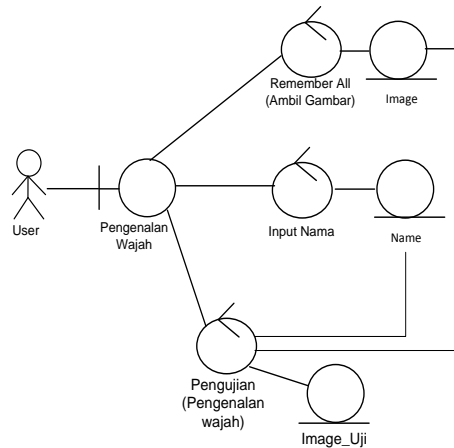
- Kondisi awal : Nama belum di masukkan
- Kondisi akhir : Nama telah di masukkan
- Skenario : *User* melakukan penginputan nama setelah pengambilan gambar selesai dilakukan dan data nama akan disimpan ke dalam sistem yang memuat data nama.

III.3.2.4 Skenario Use Case Pengujian

- Kondisi awal : *Webcam* telah di aktifkan serta data wajah dan nama telah disimpan
- Kondisi akhir : Wajah telah dikenali
- Skenario : *User* dapat melakukan pengenalan wajah dengan mengklik tombol pengujian, apabila data wajah tersebut sama dengan data wajah yang ada didalam sistem, maka sistem akan menampilkan nama *user* yang melakukan pengujian.

III.4 Analisis Kelas

Berikut ini adalah analisis kelas yang dibuat berdasarkan identifikasi objek yang ada pada skenario *use case*.



Gambar 0.3 Analisis Kelas

Gambar III.3 menjelaskan tentang analisis kelas pada aplikasi ini. Adapun rincian dari kelas-kelas pada gambar tersebut dijelaskan pada tabel III.1, seperti berikut:

Tabel 0.1. Spesifikasi Analisis Kelas

Jenis Kelas	Nama Kelas	Deskripsi
Kelas Boundary	Pengenalan Wajah	Kelas untuk menampilkan interface menu utama dari aplikasi.
Kelas Control	<i>Remember All</i>	Kelas untuk mengelola file gambar.
	<i>Input nama</i>	Kelas untuk memasukkan data nama
	Pengujian	Kelas untuk melakukan pengenalan wajah.
Kelas Entity	Image	Kelas untuk menyimpan data-data wajah.
	Name	Kelas untuk menyimpan data-data nama.
	Image Uji	Kelas untuk melakukan pengujian yang dilakukan oleh user, yang akan dicocokkan dengan data wajah yang ada didalam sistem.

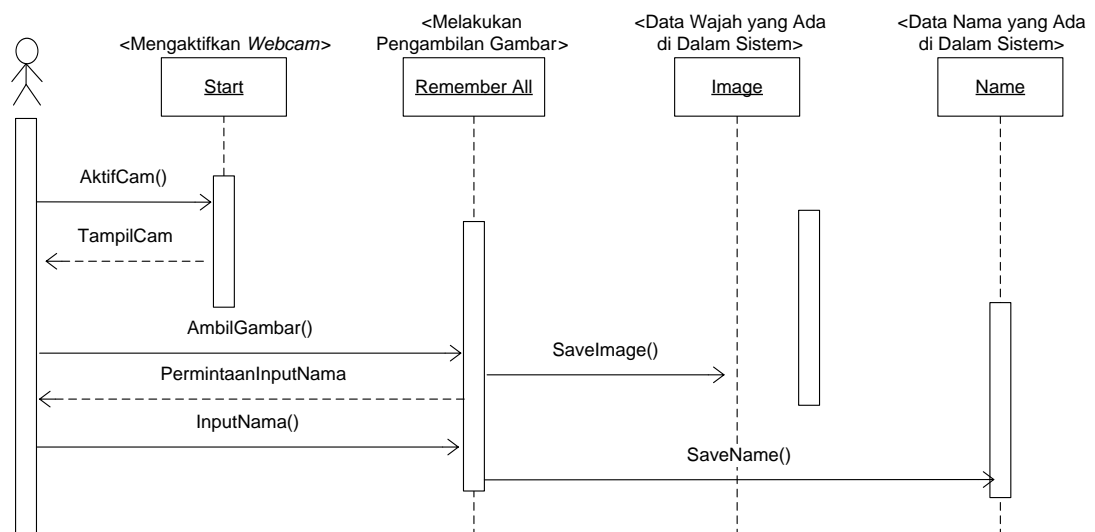
Bab IV Perancangan Sistem

IV.1 Diagram Interaksi

Interaksi Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu.

Berikut ini adalah bentuk diagram interaksi pada Aplikasi Pengenalan Wajah

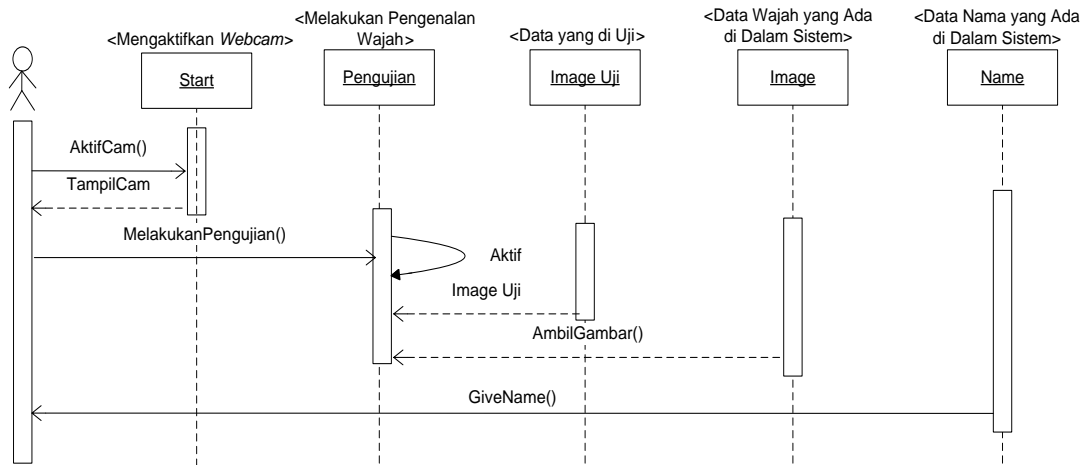
IV.1.1 Diagram Interaksi Use Case Remember All



Gambar 0.1 Diagram Interaksi Use Case Remember All

Diagram interaksi pada gambar IV.1 menjelaskan proses pengambilan gambar wajah oleh *user*. Pertama *user* harus mengaktifkan *webcam*, kemudian *user* melakukan pengambilan gambar sebanyak 10 kali per *user*. Setelah melakukan pengambilan gambar, *user* memasukkan nama lalu data gambar wajah disimpan kedalam sistem yang memuat data wajah dan data nama akan disimpan kedalam sistem yang memuat data nama.

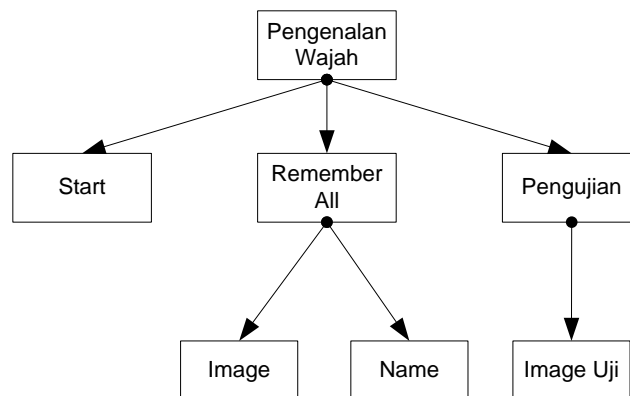
IV.1.2 Diagram Interaksi Use Case Pengujian



Gambar 0.2 Diagram Interaksi Use Case Pengujian

Diagram interaksi pada gambar IV.2 menjelaskan proses pengambilan gambar wajah oleh *user*. Pertama *user* harus mengaktifkan *webcam*, setelah itu *user* dapat melakukan pengenalan wajah dengan cara mengambil *image* uji mengklik tombol pengujian, apabila data wajah tersebut sama dengan data wajah yang ada didalam sistem yang memuat data wajah, maka sistem akan menampilkan nama *user* yang melakukan pengujian.

IV.2 Diagram Kelas



Gambar IV.3 Diagram Kelas

IV.3 Rancangan Kelas Rinci

IV.3.1 Kelas Pengenalan Wajah

Pengenalan Wajah
+Show (image)

IV.3.2 Kelas Start

Start
+StartCam (Aktiv)
+TampilLyr (Image)

IV.3.3 Kelas Remember All

Remember All
+Capture ()
+InputName ()
-username : Varchar
-facetemplate : Byte

IV.3.4 Kelas Pengujian

Pengujian
+Capture ()
+GiveName ()
-username : Varchar
-facetemplate : Byte

IV.3.5 Kelas Image

Db.Image
+Save (Image)
-facetemplate : Byte

IV.3.6 Kelas Name

Db.Name
+Save (Name)
-username : Varchar

IV.3.7 Kelas Image Uji

Db.Uji
+Save (Image)
-templateuji : Byte

IV.4 Algoritma

Bagian ini hanya diisi untuk kerangka algoritma untuk operasi/method yang dianggap cukup penting.

IV.4.1 Algoritma Pada Kelas Remember All

Nama Kelas : Remember All

Nama Operasi : Menyimpan data wajah dan data nama

Algoritma :

```
faceTemplates.Add(template);  
if (faceTemplates.Count % 10 == 0)  
{  
    InputName inputName = new InputName();  
    inputName.ShowDialog();  
    userName = inputName.userName;  
    listname.Add(userName);  
}
```

IV.4.2 Algoritma Pada Kelas Pengujian

Nama Kelas : Pengujian

Nama Operasi : Mengambil data Uji dan melakukan Pengenalan Wajah

Algoritma :

```

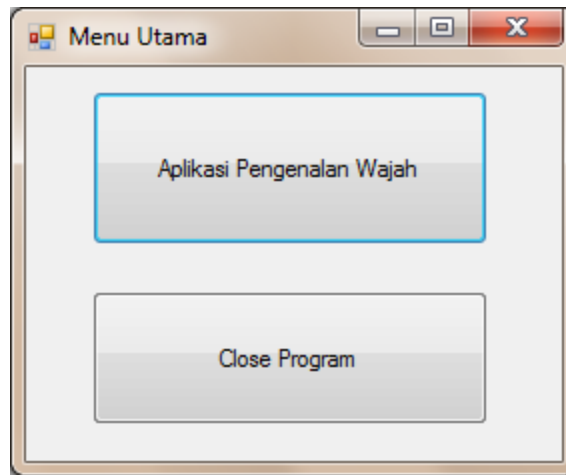
bool match = false ;
int idx = 0 ;
int idx_orang ;
foreach (FaceTemplate t in faceTemplates)
{
    float similarity = 0.0f;
    float threshold = 0.0f;
    // set FAR to 1%
    FSDK.GetMatchingThresholdAtFAR(0.01f, ref threshold);
    if (similarity > threshold)
    {
        match = true;
        break;
    }
    if (match)
    {
        idx_orang = idx / 10;
        gr.DrawString(listname[idx], new System.Drawing.Font("Arial", 16),
            break;
    }
    else
    {
        idx++;
    }
}

```

IV.5 Perancangan Antar Muka

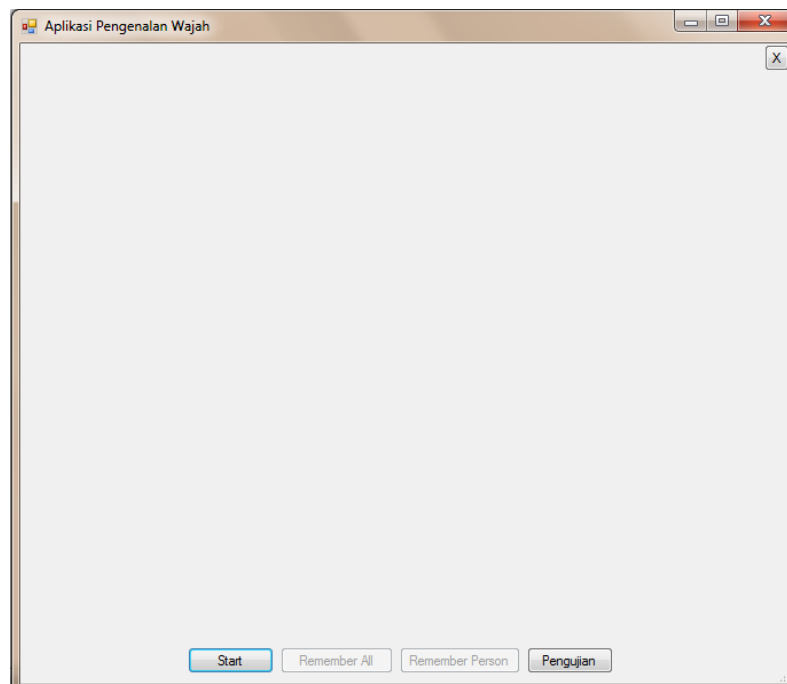
Pada bagian ini berisi tentang rancangan antarmuka untuk kelas-kelas yang diidentifikasi sebagai objek antar muka.

IV.5.1 Rancangan Tampilan Menu Utama



Gambar IV.4 Rancangan Tampilan Menu Utama

IV.5.2 Rancangan Tampilan Aplikasi Pengenalan Wajah



Gambar IV.5 Rancangan Tampilan Aplikasi Pengenalan Wajah

IV.5.3 Rancangan Tampilan Start



Gambar IV.6 Rancangan Tampilan Start

IV.5.4 Rancangan Tampilan Remember All



Gambar IV.7 Rancangan Tampilan Remember All

IV.5.5 Rancangan Tampilan Pengujian



Gambar IV.8 Rancangan Tampilan Pengujian

Bab V Implementasi dan Pengujian

Setelah dilakukan tahap perancangan maka tahap selanjutnya adalah implementasi dan pengujian pada perangkat lunak. Implementasi akan menghasilkan aplikasi yang dapat dijalankan di lingkungan operasional. Untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai deskripsi perencanaan maka perlu dilakukan pengujian.

V.1 Implementasi dan Pengujian Aplikasi

V.1.1 Implementasi Kelas

Tabel V.1 Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	Start	OpenVideoCamera.cs	OpenVideoCamera.dll
2	Remember All	psRememberAll.cs	psRememberAll.dll
3	Pengujian	psRecognize.cs	psRecognize.dll
4	Db. Image	-	-
5	Db. Name	userName.cs	userName.dll
6	Db. Uji	-	-

V.1.2 Implementasi Antarmuka

Tabel V.2 Implementasi Antarmuka

No	Antarmuka	Nama File Fisik	Nama File Executable
1	Layar Menu Utama	MenuUtama.cs	MenuUtama.dll
2	Layar Aplikasi Pengenalan Wajah	PengenalanWajah.cs	PengenalanWajah.dll
3	Layar Input Nama	InputName.cs	InputName.dll

V.2 Pengujian

V.2.1 Skenario pengujian

Pada tahapan yang dilakukan dalam pengujian ini, yang pertama dilakukan adalah mempersiapkan sebuah objek berupa wajah manusia. Objek dapat bergerak secara horizontal atau vertikal dengan pencahayaan yang baik. Jika pergerakan objek tidak horizontal atau vertikal maka ukuran objek akan terlihat mengecil ketika objek menjauh dari kamera atau membesar ketika objek mendekati kamera.

Langkah kedua, melakukan pengujian pengenalan wajah dengan 3 kondisi yaitu :

1. Melakukan pengenalan terhadap objek(wajah) yang terdeteksi oleh *webcam* dan data wajah tersebut telah ada didalam sistem.
2. Melakukan pengenalan terhadap objek(wajah) yang terdeteksi oleh *webcam* tetapi data wajah tersebut tidak ada didalam sistem.
3. Melakukan pengenalan terhadap objek(wajah) yang tidak terdeteksi oleh *webcam* tetapi data wajah tersebut ada didalam sistem.

V.2.2 Hasil Pengujian

1. Wajah terdeteksi dan data wajah ada didalam sistem



Gambar V.1 Rancangan Tampilan Pengujian 1

Gambar V.I merupakan hasil dari pengujian pertama, pengujian dilakukan terhadap objek (wajah) yang terdeteksi oleh *webcam* dan data wajah tersebut sebelumnya telah disimpan didalam sistem yang memuat data wajah, didalam sistem terdapat beberapa data wajah yaitu data wajah Prima, data wajah Fitra, data wajah Arman, dan data wajah Iwan. Dengan menggunakan metode *facematching*, sistem akan mencocokkan dan mencari kesamaan antara data yang di uji dengan data yang ada didalam sistem. Setelah sistem menemukan data tersebut sama, maka pengenalan wajah berhasil dilakukan dan aplikasi akan menampilkan nama *user* yang melakukan pengujian seperti gambar V.I.

2. Wajah terdeteksi dan data wajah tidak ada didalam sistem



Gambar V.2 Rancangan Tampilan Pengujian 2

Gambar V.2 merupakan hasil dari pengujian yang kedua, pengujian dilakukan terhadap objek (wajah) yang terdeteksi oleh *webcam*. Tetapi data wajah yang akan diuji tidak ada didalam sistem, karena sistem belum melakukan penyimpanan untuk data wajah yang akan diuji. Sehingga sistem tidak dapat menemukan data yang sama antara data uji dan data yang tersimpan didalam sistem, dan aplikasi tidak akan menampilkan nama *user* yang melakukan pengujian seperti gambar V.2.

3. Wajah tidak terdeteksi dan data wajah ada didalam sistem



Gambar V.3 Rancangan Tampilan Pengujian 3

Gambar V.3 merupakan hasil dari pengujian yang terakhir, pengujian dilakukan terhadap objek (wajah) yang tidak terdeteksi oleh *webcam*. Tetapi data wajah tersebut sebelumnya telah tersimpan didalam sistem, data wajah yang ada didalam sistem yaitu data wajah Prima, data wajah Fitra, data wajah Arman, dan data wajah Iwan. Karena wajah *user* tidak terdeteksi, maka sistem tidak dapat mengambil data wajah yang akan diuji. Sehingga aplikasi tidak akan melakukan proses apapun dan aplikasi tidak dapat melakukan pengenalan wajah seperti gambar V.3.

Dari gambar hasil pengujian diatas menunjukkan bahwa aplikasi sudah dapat berjalan dengan benar dalam melakukan pengenalan wajah. Pada pengenalan wajah yang terdeteksi dan data wajah ada didalam sistem, pengenalan wajah yang terdeteksi dan data wajah tidak ada didalam sistem dan yang terakhir adalah pengenalan wajah yang tidak terdeteksi dan data wajah ada didalam sistem.

Pengenalan wajah dilakukan secara langsung, pertama yang dilakukan adalah data wajah yang diambil akan diproses terlebih dahulu yang berfungsi untuk mempermudah proses pengenalan wajah. Proses tersebut meliputi pengkonversian data wajah dari format RGB ke *grayscale* yang berfungsi agar ukuran data yang

disimpan tidak terlalu besar, kemudian dilakukanlah proses normalisasi ukuran citra(*resize*) yang berfungsi untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses dan dilakukannya proses normalisasi pencahayaan ketika mengambil citra input dan data wajah yang diambil, dan *histogram equalization* untuk memperbaiki kualitas citra input agar dapat mempermudah proses pengenalan wajah tanpa menghilangkan informasi utamanya.

Setelah proses tersebut dilakukan maka proses selanjutnya adalah *facial feature detection* dan *eye centers detection* untuk menentukan koordinat titik-titik fitur yang ada pada wajah, setelah proses tersebut selesai dilakukan maka proses pengujian dapat dilakukan dengan menggunakan metode *face matching*.

Bab VI Kesimpulan dan Saran

VI.1 Kesimpulan

Setelah di lakukan tahap analisis, perancangan, implementasi dan pengujian pada aplikasi Pengenalan Wajah Berbasis *OpenCV*, maka kesimpulan yang didapat adalah:

1. Aplikasi dapat melakukan melakukan pendeteksian wajah melalui *webcam*.
2. Aplikasi dapat melakukan pengenalan wajah terhadap objek(wajah) terdeteksi oleh *webcam* dengan catatan data wajah tersebut telah tersimpan pada sistem.

VI.2 Saran

Penyusun menyadari masih terdapat kekurangan-kekurangan dalam aplikasi Pengenalan Wajah Berbasis *OpenCV* ini. Oleh karena itu, aplikasi ini dapat dikembangkan kembali. Adapun saran yang dapat diberikan untuk penyempurnaan dari aplikasi ini adalah :

1. Diharapkan aplikasi ini dapat menyimpan data wajah dan data nama didalam database.
2. Diharapkan aplikasi ini dapat dikembangkan untuk penyempurnaan pembuatan sistem absensi dan juga sistem keamanan.

DAFTAR PUSTAKA

1. Al Fatta, Hanif, “*Konversi Format Citra RGB ke Format Grayscale Menggunakan Visual Basic*”, (2007, Nov 24). Tersedia: <http://p3m.amikom.ac.id/p3m/51%20%20KONVERSI%20FORMAT%20CI%20TRA%20RGB%20KE%20FORMAT%20GRAYSCALE.pdf> diakses pada tanggal 04 April 2011.
2. http://lecturer.eepisits.edu/~nana/index_files/materi/Prak_Citra/Praktikum%2006%20Histogram.pdf diakses pada tanggal 05 April 2011
3. <http://luxand.com/facesdk/documentation/specifications.php> diakses pada tanggal 05 Agustus 2011.
4. <http://luxand.com/facesdk/documentation/overview.php> diakses pada tanggal 05 Agustus 2011.
5. Budiharto Widodo, “*Robotika Teori + Implementasi*”, (2010). Penerbit: Andi. Yogyakarta. Hal 130.