

# Pencatatan Kehadiran Karyawan menggunakan Metode Pengenalan Wajah

Laporan Tugas Akhir

Ridwan	3310901019
Ilham Ramadhan	3310901035
Lagiyanto	3310901042

---

Disetujui oleh: **Hilda Widyastuti M.T**  
**102020**



**POLITEKNIK NEGERI** Batam

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**POLITEKNIK NEGERI BATAM**  
**BATAM**  
**2012**

**LEMBAR PENGESAHAN**  
**PENCATATAN KEHADIRAN KARYAWAN MENGGUNAKAN**  
**METODE PENGENALAN WAJAH**

**TUGAS AKHIR**

Oleh:

NIM	Nama
3310901019	: Ridwan
3310901035	: Ilham Ramadhan
3310901042	: Lagiyanto

Diajukan dan disahkan sebagai laporan Tugas Akhir  
Di Program Studi Teknik Informatika  
Politeknik Negeri Batam

Batam, April 2012

**Pembimbing,**

**Hilda Widyastuti M.T**

**NIK.102020**

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Ridwan

Nama : 3310901019

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

### PENCATATAN KEHADIRANKARYAWANMENGUNAKAN METODE PENGENALAN WAJAH

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, April 2012

**Ridwan**  
3310901019

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Ilham Ramadhan

Nama : 3310901035

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

### PENCATATAN KEHADIRAN KARYAWAN MENGGUNAKAN METODE PENGENALAN WAJAH

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, April 2012

**Ilham Ramadhan**

3310901035

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Lagiyanto

Nama : 3310901042

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

### PENCATATAN KEHADIRAN KARYAWAN MENGGUNAKAN METODE PENGENALAN WAJAH

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, April 2012

**Lagiyanto**

3310901042

## KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa, karena berkat dan rahmathidayat serta bimbingan-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul **Pencatatan Kehadiran Karyawan menggunakan Metode Pengenalan Wajah**. Aplikasi ini dibuat untuk mempermudah pengelolaan sistem pencatatan kehadiran pada suatu perusahaan.

Dalam kesempatan ini, penulis mengucapkan terima kasih kepada :

1. Bapak Uuf Brajawidagda, MT selaku koordinator dan Dosen Pengampu Tugas Akhir.
2. Ibu Hilda Widyastuti, M.T selaku pembimbing yang juga telah membimbing dalam penyelesaian Tugas Akhir ini.
3. Kedua orang tua tercinta yang telah banyak memberikan dukungan moral, doa dan semangat.
4. Andrian Muslim, Arif Nu'man dan Syahril Idwar yang saling mendukung dalam pengerjaan Tugas Akhir.
5. Teman-teman seperjuangan IF 2009 yang memberikan dukungan semangat dan doa.
6. Orang-orang spesial, yaitu :
  - a. Ridwan

Terima kasih kepada Nita Puspita atas segala dukungan dan doanya selama penyelesaian Tugas Akhir ini.

b. Ilham Ramadhan

Terima kasih kepada Martiara Novianti atas segala bentuk dukungan, doa dan semangatnya yang selalu menemani kapanpun dan dimanapun selama penyelesaian Tugas Akhir ini.

c. Lagiyanto

Terima kasih kepada Eka Kusumawati (Kha J.CO) buat dukungan dan doanya selama penyelesaian Tugas Akhir ini.

Dalam penulisan ini, penulis mengakui bahwa masih terdapat kekurangan - kekurangan dalam penyusunannya. Oleh karena itu, penulis sangat mengharapkan bantuan dari berbagai pihak berupa kritik ataupun saran guna penyempurnaan selanjutnya. Akhir kata penulis ucapkan terima kasih, semoga laporan Tugas Akhir ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah aplikasi serupa.

Batam, April 2012

Penulis

## **ABSTRAK**

### **PENCATATAN KEHADIRAN KARYAWAN MENGGUNAKAN METODE PENGENALAN WAJAH**

Tugas akhir ini bertujuan membuat aplikasi pencatatan kehadiran karyawan yang dapat memenuhi kebutuhan pencatatan kehadiran karyawan di perusahaan. Hal ini dikarenakan metode yang *post card* pada perusahaan kurang efektif sehingga sering terjadi masalah kecurangan penitipan kartu absen, sehingga data pencatatan kehadiran menjadi tidak akurat. Metode yang digunakan untuk identifikasi Pencatatan kehadiran karyawan menggunakan biometrik wajah sebagai identitas.

Aplikasi ini dapat digunakan manager dan karyawan perusahaan. Pembuatan aplikasi ini menggunakan bahasa pemrograman C# dan DBMS yang digunakan adalah MySQL. Metode yang digunakan dalam pembuatan aplikasi ini adalah PCA (*Principal Component Analysis*) yang terdapat di dalam *libraryOpenCV*. PCA melakukan reduksi dimensi data tanpa mengurangi informasi dari data tersebut, sehingga hanya fitur-fitur wajah yang signifikan yang dibandingkan agar proses pengenalan wajah berlangsung cepat sesuai kebutuhan pencatatan kehadiran di perusahaan.

Dari hasil pengujian aplikasi yang telah dilakukan dapat disimpulkan bahwa aplikasi ini dapat mengenali wajah dengan tingkat akurasi hingga 88% pada kondisi gelap dan terang. Aplikasi telah mampu mempermudah pengelolaan data perusahaan terkait pencatatan kehadiran, penyajian data-data kehadiran dan pengelolaan jadwal kerja untuk setiap karyawan.

Kata Kunci : Pencatatan kehadiran, Pengenalan Wajah, PCA, *OpenCV*

**ABSTRACT**  
**EMPLOYEES ATTENDANCE RECORDS USING**  
**FACE RECOGNITION**

This final project aims to make application to record the employees attendance. The previous post card method is less effective, because attendance fraud problem, cause inaccurate attendance records. Our method solves the problem by identifying the presence of employees by face identity.

This application can be used manager and employees. This application is built by C # programming language and MySQL. It uses PCA (Principal Component Analysis) method from OpenCV library. PCA reduces data dimension without reducing the information from data. It compares the significant facial features, causes the fast face recognition.

This application can recognize faces with accuracy rate of up to 88% in light and dark conditions. Application has been able to help the management to record and present employees attendance, and manage work schedules for each employee.

Keywords: attendance records, face recognition, PCA, OpenCV

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN .....	ii
KATA PENGANTAR .....	v
ABSTRAK .....	vii
ABSTRACT .....	viii
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiv
Bab I   Pendahuluan.....	1
I.1    Latar Belakang .....	1
I.2    Rumusan Masalah .....	2
I.3    Batasan Masalah.....	2
I.4    Tujuan .....	2
Bab II   Tinjauan Pustaka.....	3
II.1   Pencatatan Kehadiran.....	3
II.2   Pengolahan Citra Digital .....	3
II.2.1   Dasar-dasar <i>Image</i> dan <i>Pixel</i> .....	4
II.2.2   Warna RGB ( <i>Red, Green, Blue</i> ).....	4
II.2.3   Warna Tingkat Keabuan ( <i>Grayscale</i> ).....	5
II.3   OpenCV .....	6
II.4   Face Detection.....	7
II.4.1   Integral Proyeksi.....	8
II.4.2   Metode Viola-Jones.....	9
II.5   Face Recognition.....	14
II.5.1   Euclidean Distance .....	14

II.5.2	Principal Component Analysis (PCA) .....	15
II.5.3	Algoritma Eigenface.....	15
II.5.4	Perbandingan Beberapa Metode Pengenalan Wajah.....	19
II.6	Penelitian Sebelumnya .....	20
Bab III	Analisis.....	23
III.1	Deskripsi Umum Sistem .....	23
III.2	Karakteristik Pengguna .....	24
III.3	Kebutuhan Fungsional .....	25
III.4	Kebutuhan Non Fungsional.....	25
III.5	Use Case.....	26
III.5.1	Diagram Use Case .....	26
III.6	Analisis Kelas .....	29
Bab IV	Perancangan .....	31
IV.1	Sequence Diagram .....	31
IV.1.1	Sequence Diagram Kelola Karyawan .....	31
IV.1.2	Sequence Diagram Kehadiran .....	32
IV.1.3	Sequence Diagram Membuat Jadwal.....	33
IV.1.4	Sequence Diagram Melihat Kehadiran.....	34
IV.2	Diagram Kelas.....	34
IV.3	Analisis Kebutuhan Data.....	37
IV.3.1	E-R Diagram .....	37
IV.4	Rancangan Kelas Detail .....	38
IV.4.1	Kelas Kelola Karyawan .....	38
IV.4.2	Kelas Kehadiran.....	40
IV.4.3	Kelas Jadwal .....	42

IV.4.4	Kelas Melihat Kehadiran .....	44
IV.4.5	Kelas Kontrol Wajah .....	46
IV.4.6	Kelas PCA.....	47
IV.5	Perancangan Antarmuka .....	49
IV.5.1	Kelola Karyawan .....	49
IV.5.2	Jadwal .....	50
IV.5.3	Kehadiran.....	51
IV.5.4	Melihat Kehadiran .....	52
Bab V	Implementasi dan Pengujian.....	53
V.1	Implementasi Kelas.....	53
V.2	Pengujian Pengenalan Wajah.....	54
V.2.1	Skenario Pengujian Pengenalan wajah.....	54
V.2.2	Hasil Pengujian Pengenalan wajah.....	54
V.3	Pengujian Kelola Karyawan.....	60
V.3.1	Skenario Pengujian Kelola Karyawan.....	60
V.4	Pengujian Jadwal.....	64
V.4.1	Skenario Pengujian Jadwal.....	64
V.5	Pengujian Melihat Kehadiran.....	66
V.5.1	Skenario Pengujian Melihat Kehadiran.....	66
Bab VI	Kesimpulan dan Saran.....	68
VI.1	Kesimpulan .....	68
VI.2	Saran.....	68
DAFTAR PUSTAKA	.....	70

## DAFTAR GAMBAR

Gambar 1 Gambar dengan Warna RGB.....	5
Gambar 2 Gambar dengan Warna Tingkat Keabuan (Grayscale) .....	6
Gambar 3 penggunaan integral proyeksi dalam pengolahan citra .....	8
Gambar 4 Penghitungan dalam integral.....	9
Gambar 5 Contoh Fitur yang Digunakan Viola-Jones.....	10
Gambar 6 Contoh citra integral.....	11
Gambar 7 Integral Image .....	12
Gambar 8 Cascade Classifier .....	13
Gambar 9 Rumus Eucledian Distance.....	14
Gambar 10 Contoh Penyusunan Flatvector .....	16
Gambar 11 Contoh Perhitungan Rataan Flatvector .....	17
Gambar 12 Contoh Menentukan Nilai Eigenface .....	17
Gambar 13 Perhitungan Eigenface untuk Testface .....	18
Gambar 14 Proses Identifikasi .....	18
Gambar 15 Proses Kerja Perangkat Lunak dalam Pencatatan Awal.....	23
Gambar 16 Proses Kerja Perangkat Lunak dalam Pencatatan Kehadiran.....	24
Gambar 17 Use Case Pencatatan Kehadiran Karyawan Skenario Use Case .....	26
Gambar 18 Analisis Kelas pengelolaan aplikasi .....	29
Gambar 19 Sequence Diagram Kelola karyawan .....	31
Gambar 20 Sequence Diagram Kehadiran .....	32
Gambar 21 Sequence Membuat Jadwal .....	33
Gambar 22 Sequence Melihat Kehadiran .....	34
Gambar 23 Diagram Kelas.....	36

Gambar 24 E-R Diagram .....	37
Gambar 25 Antarmuka Kelola Karyawan.....	49
Gambar 26 Antarmuka Jadwal.....	50
Gambar 27 Antarmuka Kehadiran .....	51
Gambar 28 Antarmuka Melihat Kehadiran.....	52
Gambar 29 hasil pengujian pada siang hari .....	56
Gambar 30 Hasil pengujian pada malam hari .....	59

## DAFTAR TABEL

Tabel 1 Perbandingan Metode Pengenalan Wajah.....	20
Tabel 2 Perbandingan Penelitian Sebelumnya.....	22
Tabel 3 Tabel Karakteristik Pengguna.....	25
Tabel 4 Spesifikasi Analisis Kelas.....	30
Tabel 5 Keterangan Kelas Fix.....	35
Tabel 6 Deskripsi Rancangan Antarmuka Kelola Karyawan .....	49
Tabel 7 Deskripsi Rancangan Antarmuka Jadwal .....	50
Tabel 8 Deskripsi Rancangan Antarmuka Kehadiran.....	51
Tabel 9 Deskripsi Rancangan Antarmuka Melihat Kehadiran .....	52
Tabel 10 Implementasi Kelas.....	53
Tabel 11 pengamatan I.....	54
Tabel 12 pengamatan II.....	55
Tabel 13 pengamatan III .....	55
Tabel 14 pengamatan IV .....	57
Tabel 15 pengamatan V .....	58
Tabel 16 pengamatan VI.....	58
Tabel 13 Hasil Pengujian Kelola Karyawan .....	62
Tabel 14 Hasil Pengujian Jadwal .....	65
Tabel 15 Hasil Melihat Kehadiran .....	67

## **Bab I   Pendahuluan**

### **I.1   Latar Belakang**

Karyawan adalah tenaga kerja yang dibutuhkan untuk menjalankan roda perusahaan. Karyawan dituntut untuk memenuhi tanggung jawabnya terkait dengan kewajibannya. Salah satu dari kewajiban tersebut adalah dengan hadir di tempat kerja sesuai dengan jadwal dan waktu yang telah ditetapkan oleh perusahaan tempat karyawan tersebut bekerja. Hal ini merupakan wujud dari memenuhi hak perusahaan. Dengan jumlah karyawan yang banyak, tentu pihak perusahaan membutuhkan manajemen pengelolaan karyawan yang lebih baik. Salah satu bentuk penanganan terhadap kebutuhan manajemen karyawan yang baik ialah dengan menyediakan sebuah perangkat pencatatan kehadiran yang dapat mencatat data kehadiran para karyawan.

Pizza Hut Batam merupakan salah satu perusahaan waralaba yang bergerak pada bidang restoran. Sistem pengelolaan absensi karyawan di Pizza Hut Batam masih menggunakan metode *post card*, yaitu metode kartu absensi yang dimasukkan kedalam *posting machine*, kemudian mesin akan mencetak waktu pada kartu absensi.

Akan tetapi, metode ini tidak cukup efektif bagi seorang manajer untuk melakukan perhitungan hari kerja dan kontrol disiplin waktu pada setiap karyawan yang masuk dan pulang kerja. Selain itu teknik identifikasi konvensional untuk mengenali identitas karyawan dengan menggunakan kartu absensi dinilai tidak cukup handal. Hal ini karena terdapat kemungkinan kartu absensi tersebut hilang atau kecurangan penitipan kehadiran oleh karyawan lain. Hal ini akan membuat data waktu kehadiran, keterlambatan dan jam kerja menjadi tidak akurat dan benar.

Penggunaan teknik identifikasi konvensional seperti *post card* ini semakin tergantikan oleh teknik yang lain, seperti teknik identifikasi biometrik. Teknik identifikasi biometrik didasarkan pada karakteristik alami manusia, yaitu

karakteristik fisiologis dan karakteristik perilaku sehingga tidak mudah dimanipulasi atau disalahgunakan oleh pihak lain. Salah satunya dengan metode pengenalan wajah manusia atau *face recognition*.

## **I.2 Rumusan Masalah**

Adapun rumusan masalah dari tugas akhir ini adalah dikarenakan sering terjadinya kecurangan penitipan kartu absensi (*post card*) pada sistem pencatatan kehadiran karyawan di Pizza Hut Batam, sehingga bagaimana agar karyawan wajib melakukan sendiri pencatatan kehadiran dengan identitas alami (biometrik wajah) yang tidak dapat dilakukan oleh orang lain.

## **I.3 Batasan Masalah**

Adapun batasan masalah dalam tugas akhir ini adalah :

1. Tidak menangani sistem penggajian.
2. Wajah yang akan dideteksi adalah wajah yang menghadap ke depan (frontal), dalam posisi tegak, dan tidak terhalangi sebagian oleh objek lain.

## **I.4 Tujuan**

Adapun tujuan dari tugas akhir ini adalah :

1. Mendeteksi dan mengidentifikasi wajah karyawan sebagai pengenalan identitas pencatatan kehadiran.
2. Mempermudah penyajian data-data kehadiran karyawan yang dilakukan menjadi lebih akurat dan periodik.
3. Mempermudah pengelolaan jadwal kerja untuk setiap karyawan

## **Bab II Tinjauan Pustaka**

Pada bab tinjauan pustaka akan dijelaskan mengenai Pencatatan Kehadiran, Pengolahan Citra Digital, *Open CV*, *Face Detection*, *Face Recognition* dan Penelitian Sebelumnya

### **II.1 Pencatatan Kehadiran**

Menurut Kamus Besar Bahasa Indonesia, pencatatan adalah proses, cara, perbuatan mencatat. Sedangkan kehadiran adalah adanya (seseorang, sekumpulan orang) pada suatu tempat. Jadi pencatatan kehadiran adalah perbuatan mencatat tentang adanya (seseorang, sekumpulan orang) pada suatu tempat.

Pencatatan kehadiran saat ini masih dilakukan secara manual. Salah satu cara pencatatan tersebut adalah dengan menulis kehadiran di buku catatan kehadiran karyawan. Hal ini efektif dilakukan jika individu yang hadir berjumlah sedikit. Namun pada penerapan di dunia nyata, cara ini tidak efektif ketika suatu perusahaan tersebut memiliki banyak karyawan.

Perkembangan teknologi identifikasi yang berkaitan dengan kebutuhan sistem pencatatan kehadiran semakin berkembang. Teknologi identifikasi tersebut diantaranya adalah identifikasi dengan kartu magnetik, *barcode*, sidik jari, kornea mata, *Optical Character Recognition (OCR)*, *voice recognition* (pengenalan suara), *face recognition* (pengenalan wajah), *Radio Frequency Identification (RFID)* dan lain sebagainya.

### **II.2 Pengolahan Citra Digital**

Menurut Kamus Besar Bahasa Indonesia, Citra adalah rupa, gambar atau gambaran. Pengolahan citra digital atau sering disebut *image processing* adalah suatu bentuk pengolahan atau pemrosesan data dengan *input* berupa gambar (*image*) dan ditransformasikan menjadi gambar lain sebagai keluarannya dengan teknik tertentu.

Image merupakan informasi yang secara umum tersimpan dalam bentuk pemetaan bit-bit, atau sering dikenal dengan bitmap. Setiap bit-bit membentuk satu

titik informasi yang berbentuk angka-angka yang menunjukkan besar intensitas yang disebut *pixel*. Atau dengan kata lain, satu *pixel* merupakan satu titik image yang terdiri dari satu atau beberapa bit informasi. Satuan dari *pixel* biasanya dinyatakan dengan posisi x, posisi y dan nilai dari *pixel* (warna atau gray). Dalam satu bidang gambar, sepenuhnya terdiri dari *pixel-pixel*. Karena itu, file yang menyimpan image biasanya ukurannya sangat besar. Image ini biasa disimpan dengan nama BMP. Untuk mengurangi ukuran dari file, biasanya file image dimampatkan dengan menggunakan teknik tertentu, misalkan yang terkenal JPEG atau GIF.

### II.2.1 Dasar-dasar *Image* dan *Pixel*

Satu satuan informasi terkecil dalam suatu layar monitor, televisi atau peraga lainnya yang menggambarkan atau membentuk suatu bayangan (*image*) disebut sebagai *pixel*. *Pixel* berasal dari kata *picture element*, yaitu bagian-bagian terkecil dari suatu gambar. *Pixel* dapat juga disebut sebagai titik gambar karena dalam dunia digital, gambar dibentuk dari titik-titik.

Citra dinyatakan dalam bentuk data matriks 2 dimensi, dimana setiap titik data mewakili satu *pixel*. Dalam hubungannya dengan data video, maka satu gambar (*image*) dikenal sebagai satu frame. Misalnya sebuah gambar dikatakan resolusinya sebesar 800 x 600 maka berarti panjang *pixel* horizontalnya 800 dan panjang *pixel* vertikalnya 600 dan jumlah total keseluruhan *pixel* dari gambar tersebut yaitu 480000 atau dapat dikatakan bahwa gambar tersebut terdiri dari 480000 *pixel*. Gambaran dimensimatriks yang mewakili 1 frame citra dengan biasa dinyatakan dengan ukuran W x H, dimana W adalah Weight dan H adalah Height.

### II.2.2 Warna RGB (*Red, Green, Blue*)

Setiap titik pada layar berisi angka yang bukan menunjukkan intensitas warna dari titik tersebut, melainkan menunjukkan nomor dari warna yang dipilih, dimana pada tiap titik dapat memilih sebanyak 256 warna, Jika suatu citra memiliki 256 warna, setiap komponen warna diwakili dengan satu *byte*. Jadi untuk masing-

masing komponen R (*Red*), G (*Green*), dan B(*Blue*) mempunyai variasi dari 0 sampai 255. Total variasi yang dihasilkan untuk sistem warna digital ini adalah  $256 \times 256 \times 256$  atau 16,777,216 jenis warna. Karena setiap komponen diwakili dengan satu *byte* atau delapan bit, maka total bit yang digunakan untuk merepresentasikan warna adalah  $8 + 8 + 8$  atau 24 bit.



**Gambar 1** Gambar dengan Warna RGB

Gambar 1 merupakan contoh Gambar dengan Warna RGB.

### II.2.3 Warna Tingkat Keabuan (*Grayscale*)

Kecerahan dari citra yang disimpan dengan cara pemberian nomor pada tiap-tiap *pixel*nya. Semakin tinggi nomor *pixel*nya maka makin terang (putih) *pixel* tersebut. Sedangkan semakin kecil nilai suatu *pixel*, mengakibatkan warna pada *pixel* tersebut menjadi gelap. Dalam sistem kecerahan yang umum terdapat 256 tingkat untuk setiap *pixel* skala kecerahan seperti ini dikenal sebagai *grayscale*. Proses mengubah gambar warna menjadi gambar *grayscale* digunakan dalam pengolahan citra untuk menyederhanakan model gambar. *Grayscale* adalah warna-warna *pixel* sebuah gambar yang dikonversi menjadi gambar abu-abu. Sistem *grayscale* hanya memerlukan satu byte atau delapan bit untuk menyimpan data, sehingga hanya mempunyai variasi dari 0 (hitam) sampai 255 (putih). Ada beberapa macam cara untuk mengkonversi sistem warna RGB menjadi *grayscale* yaitu :

- Dengan rata-rata setiap komponen warna RGB.
  - $Grayscale = (R+G+B)/3$
- Menggunakan nilai maksimal komponen RGB.
  - $Grayscale = Max \{R, G, B\}$
- Menggunakan sistem YUV (suatu ruangan warna yang dipakai sebagai bagian dari *color image pipeline* dalam sistem video dan fotografi digital)

dimana sistem ini digunakan didalam sistem NTSC (*National Television System(s) Committee*) yaitu sistem televisi analog yang digunakan di Amerika Serikat dan banyak negara lainnya, termasuk Amerika dan beberapa bagian Asia Timur, dengan cara mengambil komponen Y(iluminasi). Komponen Y sendiri diperoleh dari sitem warna RGB dengan konversi :

$$\text{Grayscale} = Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$



**Gambar 2 Gambar dengan Warna Tingkat Keabuan (Grayscale)**

Gambar 2 merupakan contoh dari Gambar Dengan Warna Tingkat Keabuan (*Grayscale*)

### **II.3 OpenCV**

OpenCV adalah singkatan dari *Open Computer Vision*, yaitu library-library opensource yang di khususkan untuk melakukan *image processing*. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. Library ini dibuat untuk bahasa C/C++ sebagai optimasi aplikasi realtime yang mempunyai API (*Application Programming Interface*) untuk *high level* maupun *low level* dimana terdapat fungsi-fungsi yang siap dipakai untuk *loading, saving*, akuisisi gambar dan video.

Fitur-fitur yang terdapat di dalam Library OpenCV ini adalah sebagai berikut :

- Manipulasi data gambar (alokasi memori, melepaskan memori, kopi gambar, setting serta konversi gambar)
- Image/Video I/O (Bisa menggunakan kamera yang sudah didukung oleh library ini)

- Manipulasi matrix dan vektor serta terdapat juga routines linearalgebra (products, solvers, eigenvalues, SVD)
- Image processing dasar (filtering, edge detection, pendeteksian tepi, sampling dan interpolasi, konversi warna, operasi morfologi, histograms, image pyramids)
- Analisis struktural
- Kalibrasi kamera
- Pendeteksian gerak
- Pengenalan objek
- Basic GUI (Display gambar/video, mouse/keyboard kontrol, scrollbar)
- Image Labelling (line, conic, polygon, text drawing)

## II.4 Face Detection

*Face Detection* atau pendeteksian wajah adalah proses pendeteksian dan pencarian fitur wajah pada citra kamera yang selanjutnya ditandai menjadi citra terlacak sebagai output sehingga pada tahap ini sistem mengenali pola sebagai wajah atau bukan. Proses ini dilakukan dengan menggunakan compiler Microsoft Visual C++ 6.0 dan image processing pada library OpenCV dari Intel dan digunakan dalam metode Haar, yang menggunakan pendekatan statistik (Viola and Jones, 2001) dengan fitur Haar. Pelatihan dengan model statistik Haar ini menggunakan sampel positif yang memiliki fitur wajah dan sampel negatif yang dimana sampel negatif tersebut tidak memiliki fitur wajah. Kedua jenis sampel tersebut dicoba bersamaan dan perbedaannya digunakan sebagai parameter klasifikasi obyek yang terdeteksi sebagai wajah atau tidak. Informasi ini disimpan dan dikompresi sebagai parameter model statistik. Parameter klasifikasi model statistik ini dikenal dengan *a cascade of boosted tree classifier*. Proses pelatihan ini dikenal dengan algoritma *Haartraining* yang pada akhir pelatihan akan menghasilkan file berekstensi .JPG yang menjadi parameter model statistik.

### II.4.1 Integral Proyeksi

Integral proyeksi adalah suatu metode yang digunakan untuk mencari daerah lokasi dari obyek. Metode ini dapat kita gunakan untuk mendeteksi batas dari daerah gambar yang berbeda sehingga kita bisa mencari daerah lokasi kepala, tangan, kaki dan fitur – fitur lainnya. Integral ini juga di sebut dengan integral baris dan kolom dari pixel, karena integral ini menjumlahkan pixel per baris dan pixel per kolom. Dari metode ini akan dengan mudah untuk menemukan daerah lokasi objek yang diperlukan.

Rumus dari Integral Proyeksi adalah sebagai berikut :

$$h(j) = \sum_{i=1}^{Nbaris} x(i, j)$$
$$h(i) = \sum_{j=1}^{Nkolom} x(i, j)$$

Contoh penggunaan integral proyeksi dalam pengolahan citra:

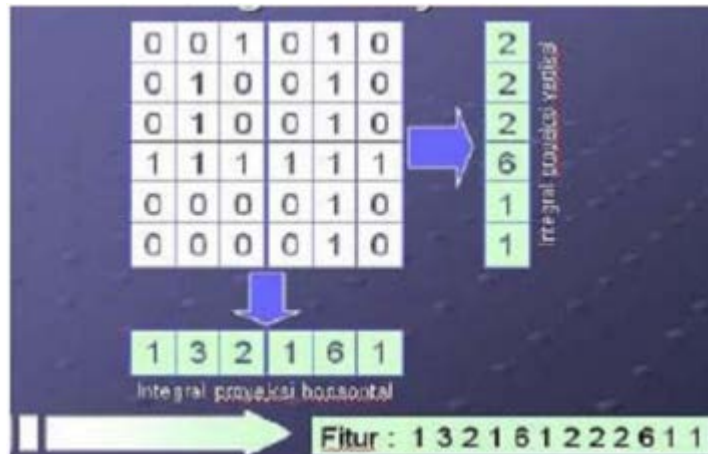


Gambar Awal

Gambar setelah integral proyeksi

Gambar 3 penggunaan integral proyeksi dalam pengolahan citra

Contoh Penghitungan dalam integral proyeksi :



Gambar 4 Penghitungan dalam integral

## II.4.2 Metode Viola-Jones

Proses deteksi adanya citra wajah dalam suatu gambar pada EmguCV, menggunakan sebuah metode yang dipublikasikan oleh Paul Viola dan Michael Jones tahun 2001. Umumnya disebut metode Viola-Jones. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama:

- Fitur segi empat sederhana yang disebut fitur Haar
- Integral image untuk pendeteksian fitur secara cepat
- Metode machine learning AdaBoost
- Pengklasifikasi bertingkat (*Cascade Classifier*) untuk menghubungkan banyak fitur secara efisien

### II.4.2.1 Fitur Haar

Proses pendeteksian wajah dilakukan dengan mengklasifikasikan sebuah gambar setelah sebelumnya sebuah pengklasifikasi dibentuk dari data latih. Data latih yang digunakan oleh algoritma ini berjumlah 5000 citra wajah dan 9400 citra non-wajah sehingga menghasilkan akurasi sistem sebesar 95% dengan data positif salah sebesar 1 : 14084.

Metode ini memerlukan 2 tipe gambar objek dalam proses *training* yang dilakukan, yaitu :

1. *Positive samples*

Berisi gambar obyek yang ingin di deteksi, apabila ingin mendeteksi wajah maka *positive samples* ini berisi gambar wajah, begitu juga obyek lain yang ingin dikenali.

2. *Negative samples*

Berisi gambar obyek selain obyek yang ingin dikenali, umumnya berupa gambar *background* (tembok, pemandangan, lantai, dan gambar lainnya). Resolusi untuk sampel negatif disarankan untuk memiliki resolusi yang sama dengan resolusi kamera. *Training* dari Haar menggunakan dua tipe sampel diatas. Informasi dari hasil training ini lalu dikonversi menjadi sebuah parameter model statistik.

Klasifikasi citra dilakukan berdasarkan nilai dari sebuah fitur. Penggunaan fitur dilakukan karena pemrosesan fitur berlangsung lebih cepat dibandingkan pemrosesan citra perpixel. Terdapat 3 jenis fitur berdasarkan jumlah persegi panjang yang terdapat di dalamnya, seperti yang dapat dilihat pada gambar 3 Contoh Fitur yang Digunakan Viola-Jones berikut :



**Gambar 5**Contoh Fitur yang Digunakan Viola-Jones

Fitur yang digunakan oleh Viola dan Jones didasarkan pada Wavelet Haar. Wavelet Haar adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah ). Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik disebut fitur Haar, atau fitur Haarlike, seperti pada gambar 5 Contoh

Fitur yang Digunakan Viola-Jones menunjukkan fitur yang digunakan dalam pendektasian citra wajah oleh Viola dan Jones.

Adanya fitur Haar ditentukan dengan cara mengurangi rata-rata pixel pada daerah gelap dari rata-rata pixel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau treshold, maka dapat dikatakan bahwa fitur tersebut ada. Nilai dari *Haar-like feature* adalah perbedaan antara jumlah nilai-nilai piksel *gray level* dalam daerah kotak hitam dan daerah kotak putih:

$$f(x) = \text{Sumblack rectangle} - \text{Sumwhite rectangle}$$

Untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien, Viola dan Jones menggunakan satu teknik yang disebut Integral Image.

### II.4.2.2 Intergral Image

Integral Image adalah sebuah citra yang nilai tiap pikselnya merupakan akumulasi dari nilai piksel atas dan kirinya. Sebagai contoh, piksel (a,b) memiliki nilai akumulatif untuk semua piksel (x,y) dimana  $x \leq a$  dan  $y \leq b$ . Contoh citra integral dapat dilihat pada gambar 6

1	2	3
4	5	6
7	8	9

Citra Masukan

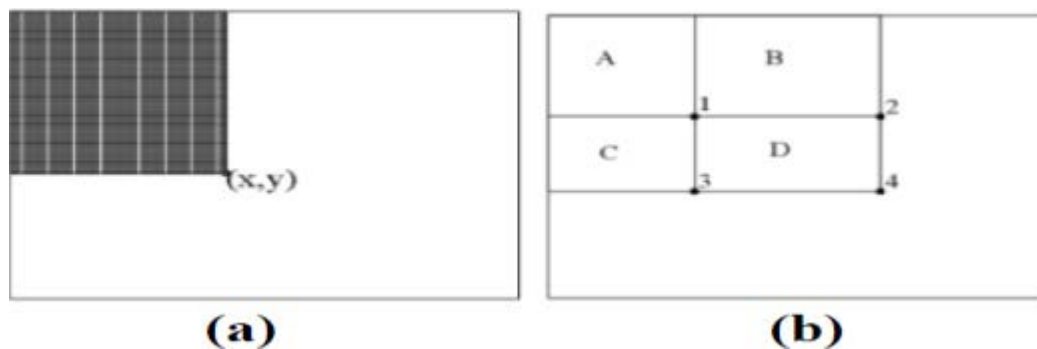
1	3	6
5	12	21
12	27	45

Citra Integral

Gambar 6 Contoh citra integral

*Integral Image* digunakan untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien. Pada umumnya, pengintegrasian tersebut berarti menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai integral untuk

masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, keseluruhan gambar itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel. Contoh integral Image dapat dilihat pada gambar 7



**Gambar 7 Integral Image**

Seperti yang ditunjukkan oleh gambar 4(a) di atas setelah pengintegrasian, nilai pada lokasi pixel (x,y) berisi jumlah dari semua pixel di dalam daerah segiempat dari kiri atas sampai pada lokasi (x,y) atau daerah yang diarsir. Untuk menentukan nilai rata-rata pixel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada (x,y) oleh area segiempat.

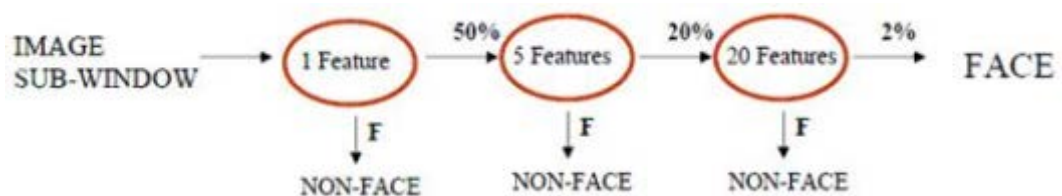
Untuk mengetahui nilai pixel untuk beberapa segiempat yang lain missal, seperti segiempat D pada gambar 4(b) di atas dapat dilakukan dengan cara menggabungkan jumlah pixel pada area segiempat  $A+B+C+D$ , dikurangi jumlah dalam segiempat  $A+B$  dan  $A+C$ , ditambah jumlah pixel di dalam A. Dengan,  $A+B+C+D$  adalah nilai dari integral image pada lokasi 4,  $A+B$  adalah nilai pada lokasi 2,  $A+C$  adalah nilai pada lokasi 3, dan A pada lokasi 1. Sehingga hasil dari D dapat dikomputasikan  $D = (A+B+C+D)-(A+B)-(A+C)+A$ .

Untuk memilih fitur Haar yang spesifik yang akan digunakan dan untuk mengatur nilai ambangnya (threshold), Viola dan Jones menggunakan sebuah metode machine learning yang disebut AdaBoost.

### II.4.2.3 Algoritma AdaBoost (Cascade Classifier)

AdaBoost menggabungkan banyak classifier lemah untuk membuat sebuah classifier kuat. Lemah disini berarti urutan filter pada classifier hanya mendapatkan jawaban benar lebih sedikit. Jika keseluruhan classifier lemah digabungkan maka akan menjadi classifier yang lebih kuat. AdaBoost memilih sejumlah classifier lemah untuk disatukan dan menambahkan bobot pada setiap classifier, sehingga akan menjadi classifier yang kuat.

Viola Jones menggabungkan beberapa AdaBoost classifier sebagai rangkaian filter yang cukup efisien untuk menggolongkan daerah image. Masing-masing filter adalah satu AdaBoost classifier terpisah yang terdiri classifier lemah atau satu filter Haar.



Gambar 8 Cascade Classifier

Threshold yang dapat diterima untuk masing-masing level filter di set rendah. Selama proses pemfilteran, bila ada salah satu filter gagal untuk melewati sebuah daerah gambar, maka daerah itu langsung digolongkan sebagai bukan wajah. Namun ketika filter melewati sebuah daerah gambar dan sampai melewati semua proses filter yang ada dalam rangkaian filter, maka daerah gambar tersebut digolongkan sebagai wajah. Viola dan Jones memberi sebutan cascade. Urutan filter pada cascade ditentukan oleh bobot yang diberikan AdaBoost. Filter dengan bobot paling besar diletakkan pada proses pertama kali, bertujuan untuk menghapus daerah gambar bukan wajah secepat mungkin.

Pada klasifikasi tingkat pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Klasifikasi ini kira-kira akan menyisakan 50% subcitra untuk diklasifikasi di tahap kedua. Seiring dengan bertambahnya tingkatan klasifikasi,

maka diperlukan syarat yang lebih spesifik sehingga fitur yang digunakan menjadi lebih banyak. Jumlah subcitra yang lolos klasifikasi pun akan berkurang hingga mencapai jumlah sekitar 2%.

## II.5 Face Recognition

*Face recognition* adalah aplikasi komputer yang untuk mengidentifikasi atau memverifikasi seseorang dari gambar digital atau *frame* video dari sumber video. Salah satu cara untuk melakukan ini adalah dengan membandingkan fitur wajah yang dipilih dari gambar dan database wajah.

Dalam sistem pengenalan wajah dibutuhkan sebuah cara untuk menangkap ciri dari setiap wajah. Ada banyak teknik untuk membandingkan fitur wajah dengan database wajah, salah satunya dengan metode *Principal Component Analysis* (PCA) yang kemudian digunakan untuk klasifikasi identitas citra dan Euclidean Distance untuk mencari kemiripan satu citra dari kelompok klasifikasi citra yang ada.

### II.5.1 Euclidean Distance

Metode yang digunakan adalah *Euclidean Distance* dikenal juga dengan perhitungan jarak satu suatu data terhadap sekelompok data (data set dari *database*). Metode ini sederhana dan cepat sehingga sering digunakan dalam berbagai sistem deteksi, identifikasi, verifikasi atau klasifikasi. Metode ini juga digunakan dalam *Principal Component Analysis*.

Di dalam metode *Euclidean Distance* semakin besar nilai disimilaritas, maka semakin besar perbedaan kedua warna tersebut, sebaliknya, semakin kecil (mendekati nol), maka semakin mirip kedua warna tersebut.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \dots + (n_2 - n_1)^2}$$

Gambar 9 Rumus Euclidean Distance

$x_1, y_1, \dots, n_1$  adalah tingkat keabuan untuk sebuah *pixel* dalam tiap kanal sedangkan  $x_2, y_2, \dots, n_2$  adalah *mean* dari masing-masing kelas pada *trainingsample*. *Pixelyang* sedang dihitung akan dimasukkan pada kelas dengan nilai *Euclidean Distance* terkecil.

## II.5.2 Principal Component Analysis (PCA)

Secara umum, *Principal Component Analysis* (PCA) dapat diartikan sebagai suatu cara untuk pengenalan pola terhadap suatu kumpulan data dan mengekspresikan kumpulan data tersebut terhadap kesamaan dan perbedaan diantaranya. PCA dapat digunakan untuk melakukan reduksi dimensi data tanpa mengurangi informasi-informasi dari data tersebut. Dalam metode ini citra wajah akan diproyeksi menjadi sebuah ruang fitur yang menonjolkan variasi yang signifikan diantara citra yang diketahui. Fitur signifikan inilah sering disebut nilai dari *Eigenface*, karena fitur-fitur tersebut adalah komponen utama dari set citra wajah untuk pelatihan. Tujuan PCA adalah menangkap variasi total pada citra latih dan merepresentasikan variasi tersebut dalam variabel-variabel yang jumlahnya lebih sedikit.

## II.5.3 Algoritma Eigenface

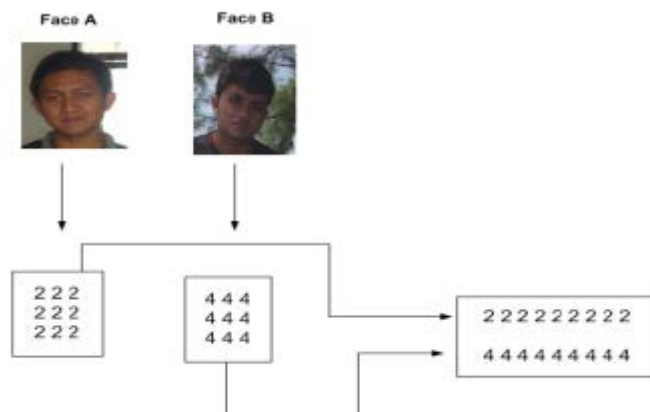
Algoritma *eigenface* adalah salah satu algoritma pengenalan wajah yang berdasarkan pada *Principal Component Analysis* (PCA) yang dikembangkan di MIT. Untuk menghasilkan *eigenface*, sekumpulan besar citra digital dari wajah manusia diambil pada kondisi pencahayaan yang sama dan kemudian dinormalisasikan dan kemudiandiolah pada resolusi yang sama (misalnya  $m \times n$ ), dan kemudian diperlakukan sebagai *vector* dimensi  $mn$  yang komponennya diambil dari nilai *pixel*-nya. Algoritma *eigenface* secara keseluruhan cukup sederhana. Citra pelatihan direpresentasikan dalam sebuah vektor flat (gabungan vektor) dan digabung bersama-sama menjadi sebuah matriks tunggal yang disebut *Eigenvector*.

*Eigenvector* kemudian diekstraksi dan disimpan dalam *file temporary* atau

*database*. Gambar pelatihan kemudian kemudian diproyeksikan dalam *feature space* (sebuah ruang abstrak dimana setiap bagian dari sampel direpresentasikan sebagai sebuah *point*), yang dinamai *face space* yang ditentukan oleh eigenvektor. Langkah-langkah pengenalan wajah dengan algoritma *eigenface* adalah sebagai berikut:

1. Penyusunan *Flatvector*

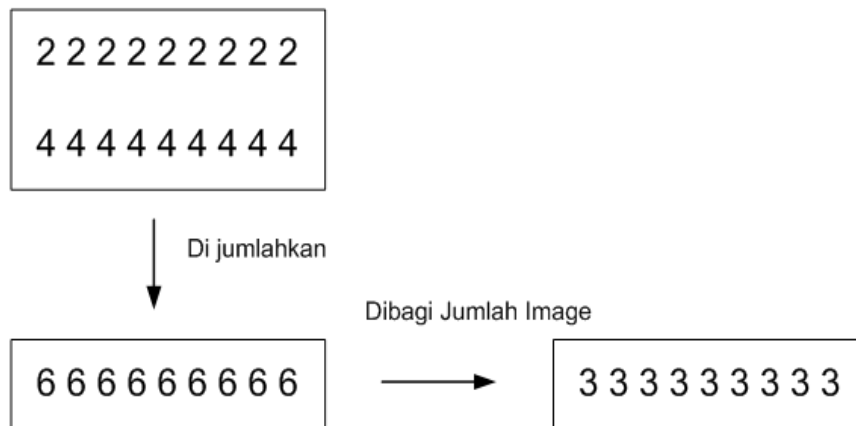
Langkah pertama adalah menyusun sebuah gambar pelatihan menjadi 1 matriks tunggal. Misalnya image yang disimpan berukuran  $H \times W$  *pixel* dan jumlahnya  $N$  buah, maka akan memiliki *flatvector* dengan dimensi  $N \times (H \times W)$ , dimana  $H$  adalah Height,  $W$  adalah Weight dan  $N$  adalah banyaknya gambar. Contoh gambaran penyusunan *flatvector* dapat dilihat pada Gambar 3 Contoh Penyusunan *Flatvector*.



**Gambar 10 Contoh Penyusunan Flatvector**

2. Perhitungan Rataan *Flatvector*

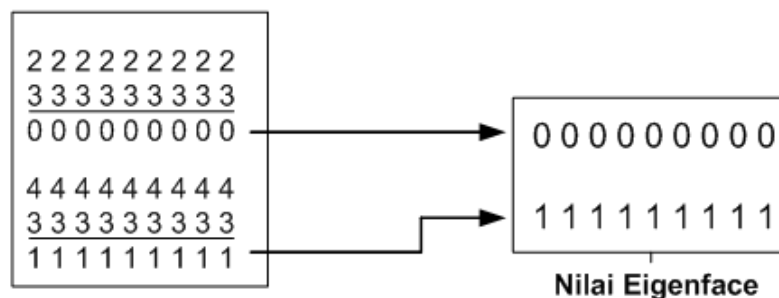
Dari *flatvector* yang diperoleh, jumlahkan seluruh barisnya sehingga diperoleh matriks tunggal berukuran  $1 \times (H \times W)$ . Setelah itu bagi matriks tadi dengan jumlah image  $N$  untuk mendapatkan rata-rata *flatvector*. Contoh Perhitungan Rataan *flatvector* dapat dilihat pada Gambar 4 Contoh Perhitungan Rataan *Flatvector*.



Gambar 11 Contoh Perhitungan Rataan Flatvector

3. Tentukan Nilai *Eigenface*

Dengan menggunakan rata-rata *flatvector* akan dihitung *eigenface* untuk matriks *flatvector* yang telah disusun. Caranya dengan mengurangi baris-baris pada matriks *flatvector* dengan rata-rata *flatvector*. Jika didapatkan nilai di bawah nol, ganti nilainya dengan nol. Contoh gambaran menentukan nilai *eigenface* dapat dilihat pada Gambar 5 Contoh Menentukan Nilai *Eigenface*.



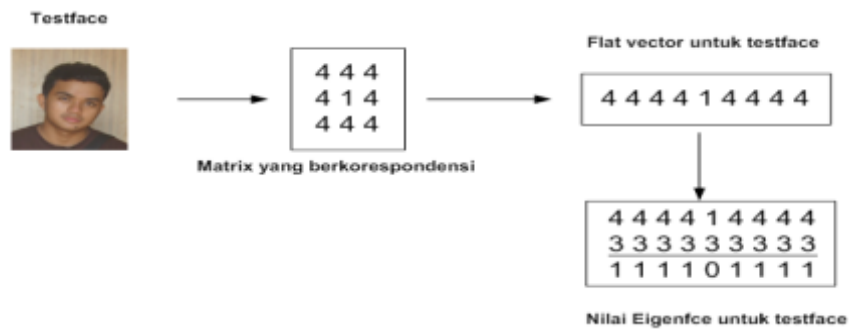
Gambar 12 Contoh Menentukan Nilai Eigenface

4. Proses Identifikasi

Jika diberikan citra yang akan diidentifikasi (*testface*), maka langkah identifikasinya adalah sebagai berikut:

Kalkulasi nilai *eigenface* untuk matriks *testface*, dengan cara yang sama dengan penentuan *eigenface* untuk *flatvector*. Gambaran perhitungan *eigenface* untuk *testface* dapat dilihat pada Gambar 6 Perhitungan

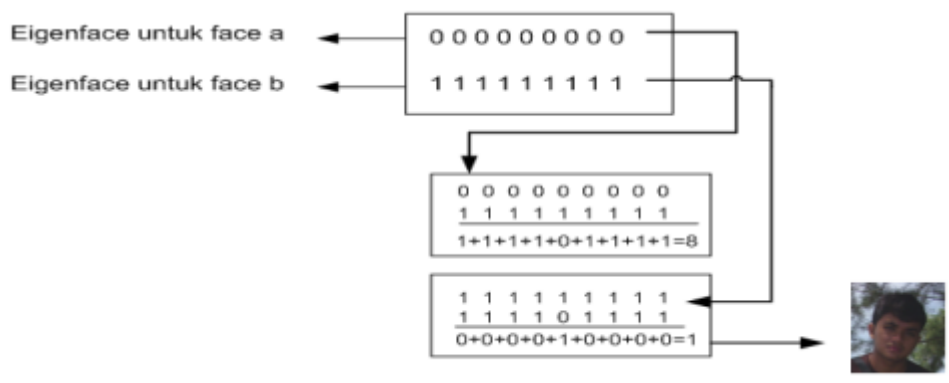
*Eigenface* untuk *Testface*.



**Gambar 13 Perhitungan Eigenface untuk Testface**

Setelah nilai *eigenface* untuk *testface* diperoleh maka dapat dilakukan identifikasi dengan menentukan jarak (*distance*) terpendek dengan *eigenface* dari *eigenvector* gambar pelatihan.

Caranya tentukan nilai absolute dari penggunaan baris *i* pada matrik *eigenface* gambar pelatihan dengan *eigenface* dari *testface*, kemudian jumlahkan elemen-elemen penyusun *vector* yang dihasilkan dari pengurangan tadi dan ditemukan jarak *d* indeks *i*. Lakukan untuk semua baris dan cari nilai *d* yang paling kecil. Gambaran proses identifikasi dapat dilihat pada Gambar 7 Proses Identifikasi.



**Gambar 14 Proses Identifikasi**

Pada proses pengenalan dengan openCV tingkat threshold (jarak ambang) konstanta kemiripan kriteria wajah ditetapkan dengan nilai eigen threshold lebih besar dari 0.001 setiap kali wajah test dicocokkan dengan wajah pelatihan. Jika nilai eigen wajah test lebih kecil dari 0.001 maka wajah test dianggap tidak ada.

Jika nilai eigen wajah test lebih besar sama dengan 0.001 maka wajah pelatihan akan dideteksi mirip dengan wajah test .

#### **II.5.4 Perbandingan Beberapa Metode Pengenalan Wajah**

Ada beberapa macam metode pengenalan wajah , yaitu *fractal neighbour distance*, *template matching*, dan *Principal Component Analysis (PCA)*. *Fractal neighbour distance* adalah metode yang menyamakan sebuah gambar yang belum teridentifikasi dengan salah satu gambar yang ada pada *database*. Metode *template matching* adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian-bagian kecil dari gambar yang cocok dengan template gambar. Sedangkan metode *Principal Component Analysis (PCA)* adalah suatu cara untuk pengenalan pola terhadap suatu kumpulan data dan mengekspresikan kumpulan data tersebut terhadap kesamaan dan perbedaan diantaranya. Metode-metode tersebut memiliki kelebihan dan kekurangan masing-masing. Namun dari penelitian yang sebelumnya ketiga metode tersebut memiliki keakuratan yang berbeda-beda. Dalam thesis Hossein Ebrahimpour yang berjudul “*Fractal Techniques for Face Recognize*” tingkat akurasi *Fractal neighbour distance* hingga 95%, *template machine* berdasarkan penelitian Mohamad Aditya Rahman pada makalah “Sistem Pengenalan Wajah dengan Metode *Template Matching*”, didapat kesesuaian keseluruhan data pada pencahayaan normal, tetapi sistem error pada pencahayaan tinggi, sehingga rata-rata keakuratan pengenalan pola wajah 60%, dan berdasarkan penelitian yang dilakukan Setya Bayu, Ahmad Hendriawan dan Ronny Susetyoko pada Penerapan Face Recognition dengan Metode *Eigenface* dalam *Intellegent Securty Home* tingkat akurasi hingga 87,5 % dengan kecepatan rata-rata 12,5 fps.

Perbandingan metode-metode tersebut dapat dilihat pada tabel 1 Perbandingan metode pengenalan wajah :

**Tabel 1 Perbandingan Metode Pengenalan Wajah**

	<i>Fractal Neighbour Distance</i>	<i>Template Matching</i>	<i>Principal Component Analysis (PCA)</i>
Kelebihan	FND memberikan hasil yang memiliki tingkat akurasi yang tinggi dalam proses identifikasinya karena ia membandingkan informasi byte pixel secara fractal, yaitu himpunan matematis yang menunjukkan kesamaan diri sendiri terhadap semua skala pembesaran., sehingga bisa didapat tingkat akurasi yang tinggi dan tidak bergantung gambar awal proses.	Prosedur pengenalan pola yang sederhana yang didasarkan pada ketepatan konfigurasi informasi penginderaan dengan konfigurasi pada <i>database</i> .	Sistem pengurangan dimensi yang menyebabkan proses pengenalan dapat berjalan dengan sangat cepat
Kekurangan	Proses pemindaian objek fractal ini cukup panjang. karena harus melalui transformasi gambar yang tergeneralisasi dengan dirinya sendiri secara berulang-ulang terus-menerus untuk menghasilkan gambar yang mendekati gambar asli. Sehingga proses berjalan lambat.	Jika perbandingan eksternal objek dengan internal objek 1:1, maka objek yang berbeda sedikit saja dengan template tidak akan dikenali. Faktor cahaya tinggi juga dapat menyebabkan sistem error, Untuk itu dibutuhkan jutaan template yang spesifik perlu dibuat dengan kondisi standar, agar cocok dengan bentuk geometri yang ada di <i>database</i>	Ketergantungan terhadap pergeseran dan intensitas cahaya

## II.6 Penelitian Sebelumnya

Terdapat dua aplikasi Tugas Akhir yang memiliki persamaan dan dijadikan sebagai referensi dalam pengerjaan Tugas Akhir ini:

- Fauzan. (3310701082) dengan judul : Aplikasi Perhitungan absensi RFID.
- Valentine (33106039) dengan judul : Aplikasi *Face recognition*

- Swara Adi Cakti Prima Dasa (3310801114) dan Fitra Fajrianto (3310801116) dengan judul : Aplikasi Pengenalan Wajah Berbasis OpenCV

Ada beberapa persamaan dan perbedaan antara masing-masing Tugas Akhir tersebut. Hal tersebut akan dijelaskan dalam Tabel 2 Perbandingan Penelitian Sebelumnya

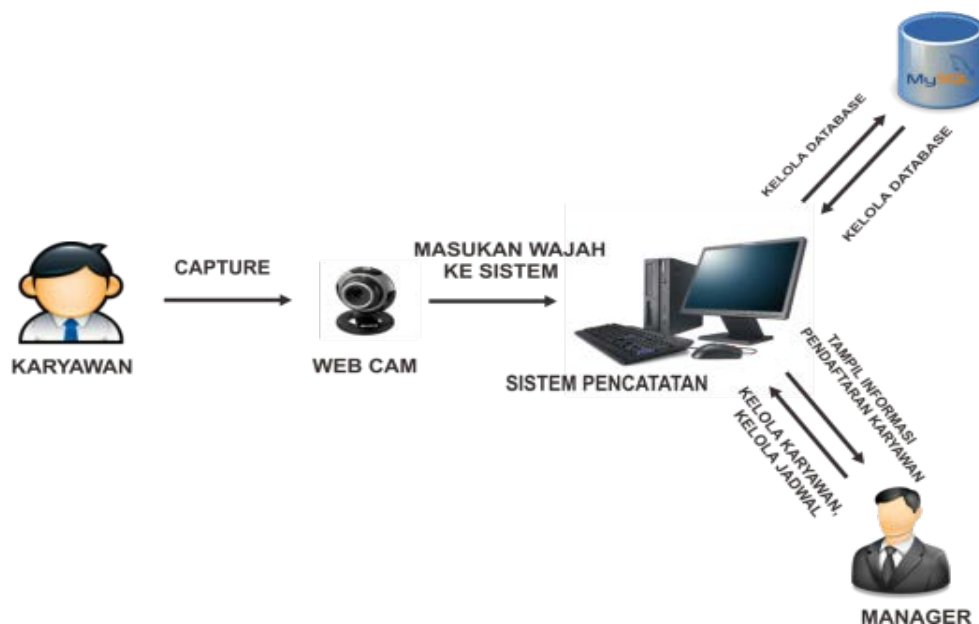
Tabel 2 Perbandingan Penelitian Sebelumnya

<b>Perbandingan</b>	<b>Aplikasi Perhitungan absensi RFID</b>	<b>Aplikasi pengenalan wajah berbasis OpenCV</b>	<b>Sistem absensi di Pizza Hut Batam</b>	<b>Pencatatan Kehadiran Karyawan menggunakan Metode Pengenalan Wajah</b>
Pengolahan absensi	Ditangani	Tidak Menangani	Ditangani	Ditangani
Hardware yang digunakan	RFID	Webcam	Posting Machine	Webcam
<i>Face recognition</i>	Tidak Menangani	Ditangani	Tidak Menangani	Ditangani
Penggajian	Ditangani	Tidak Menangani	Tidak Menangani	Tidak Menangani
Studi kasus	Matahari Dept. Store	-	Pizza Hut Batam	Pizza Hut Batam
Bahasa pemrograman	C#	C#	-	C#
Algoritma acuan	-	Tempalte Matching	-	PCA(PrincipalComponenten Analytical)/ <i>Eigenface</i>
Metode Pemrograman	Prosedural	OOP	-	OOP
Sistem Absensi	Ditangani	Tidak Menangani	Ditangani	Ditangani

## Bab III Analisis

Bab analisis ini akan menjelaskan mengenai Deskripsi Umum Sistem, Karakteristik Pengguna, Fitur Utama Perangkat Lunak, Kebutuhan Fungsional, Kebutuhan Non Fungsional, Diagram Use Case dan Analisis Kelas.

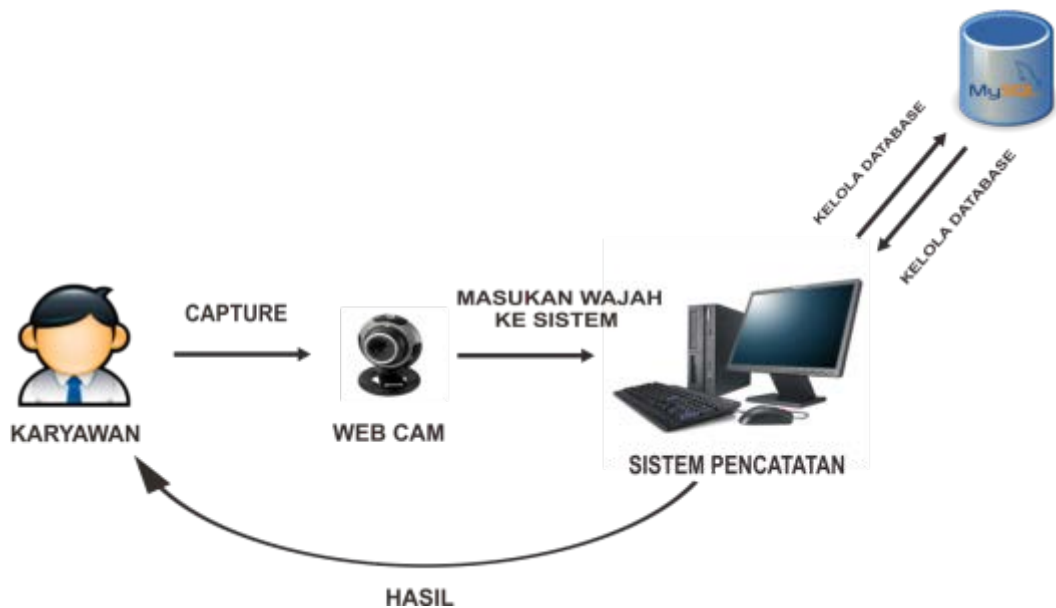
### III.1 Deskripsi Umum Sistem



Gambar 15 Proses Kerja Perangkat Lunak dalam Pencatatan Awal

Keterangan :

1. Karyawan melakukan pencatatan awal untuk pendaftaran data karyawan tersebut.
2. Karyawan melakukan sesi foto untuk data karyawan tersebut kemudian menambahkan atribut karyawan dengan NIK (Nomor Induk Karyawan), Nama\_K, Jabatan pada system
3. Wajah karyawan akan di-capture sebanyak sepuluh kali kemudian disimpan dengan identitas yang sesuai untuk karyawan tersebut
4. Semua hasil dari pencatatan awal kemudian dimasukkan kedalam database karyawan
5. Manajer dapat melakukan kelola karyawan serta melihat data karyawan yang telah telah disimpan pada saat pencatatan awal



**Gambar 16 Proses Kerja Perangkat Lunak dalam Pencatatan Kehadiran**

Keterangan :

1. Karyawan yang hadir akan melakukan pencatatan kehadiran sebagai data bahwa karyawan tersebut hadir
2. Karyawan akan melalui sesi foto dimana hasil dari foto karyawan tersebut kemudian dicocokkan dengan data-data yang terdapat didalam database
3. Setelah data karyawan tersebut didapatkan, maka karyawan dapat melihat pemberitahuan bahwa proses pencatatan kehadirann telah berjalan dan data karyawan tersebut telah tercatat dalam daftar hadir

Dalam penggambaran aplikasi Pencatatan Kehadiran Karyawan Dengan Metode Pengenalan Wajah ini terdapat fitur utama yaitu diantaranya :

1. Proses Daftar
2. Proses Capture wajah
3. Proses Pencatatan Karyawan
4. Proses penyimpanan Data Ke Database

### **III.2 Karakteristik Pengguna**

Dalam Tabel 3 merupakan karakteristik pengguna yang berperan dalam kegiatan pencatatan kehadiran karyawan

**Tabel 3 Tabel Karakteristik Pengguna**

<b>Pengguna</b>	<b>Tugas</b>	<b>Hak Akses</b>	<b>Jabatan</b>
Karyawan	Mendaftarkan diri sebagai karyawan, mencatat kehadiran dan melihat data karyawan tersebut	<i>View</i> dan <i>Insert</i> data(wajah)	Karyawan Pizza
Admin	Mendaftarkan karyawan, menambah dan mengurangi data karyawan serta melihat data karyawan	<i>Insert</i> data, <i>Update</i> data, <i>View</i> data	Manajer Pizza

### **III.3 Kebutuhan Fungsional**

Kebutuhan fungsional diawali huruf F diikuti dengan nomor urut dan fungsionalnya. Dibawah ini merupakan penjelasan kebutuhan fungsional:

F-001 Sistem menerima masukan data gambar wajah dari karyawan

F-002 Sistem mampu mencari dan mengolongkan data yang memiliki kemiripan

F-003 Sistem menyediakan fitur menambah dan menghapus karyawan

F-004 Sistem menyediakan fitur pengelolaan jadwal karyawan

F-005 Sistem menampilkan informasi pencatatan kehadiran karyawan

### **III.4 Kebutuhan Non Fungsional**

Kebutuhan non fungsional diawali huruf NF (Nilai Fungsional) yang diikuti dengan nomor dan fungsionalnya.

Dibawah ini merupakan penjelasan kebutuhan non fungsional

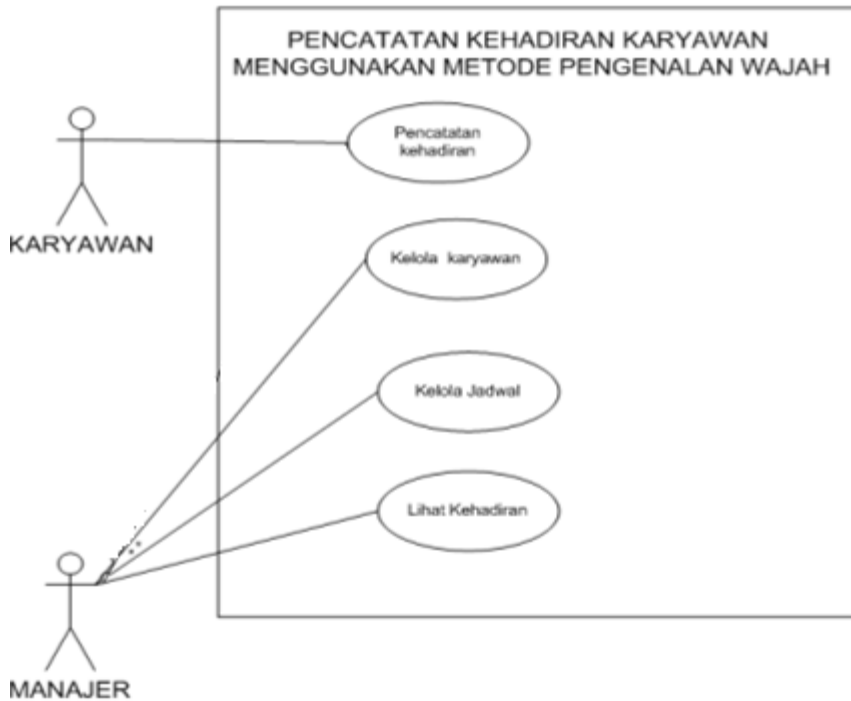
NF-001 Sistem Pencatatan Kehadiran karyawan menggunakan sistem operasi Windows

NF-002 Sistem menggunakan kamera minimal webcam IMP untuk terdeteksi

NF-003 Sistem memerlukan penerangan yang cukup pada malam hari

### III.5 Use Case

#### III.5.1 Diagram Use Case



Gambar17 Use Case Pencatatan Kehadiran Karyawan Skenario Use Case

Skenario pada aplikasi Pencatatan Kehadiran Karyawan Menggunakan Metode Wajah adalah karyawan melakukan pendaftaran awal yang kemudian data akan diproses dan disimpan ke dalam database. Kemudian karyawan mendapatkan hak akses untuk melakukan pencatatan kehadiran.

##### III.5.1.1 Kelola Karyawan

- Aktor : Manajer dan Karyawan
- Kondisi Awal : - Sistem pencatatan kehadiran telah aktif  
- Data karyawan belum terdaftar didalam database
- Kondisi Akhir : - Informasi data karyawan telah diberi label  
- Informasi karyawan disimpan dalam database

- Informasi label karyawan yang diubah akan berubah didalam database
  - Informasi label karyawan yang dihapus akan hilang dari database
- Skenario :
- Pencatatan awal :
- Karyawan atau Manajer mencapture wajahnya di depan webcam dengan kondisi frontal. Jika wajah terdeteksi bagian wajah akan di crop dengan resolusi 60x60, kemudian di *convert* ke grayscale.
  - Hasil capture kemudian akan dimasukkan kedalam database
  - Manajer mengisi form daftar karyawan dengan memberikan label nama, jabatan, untuk setiap karyawan. Informasi data akan di ambil sebanyak 10 kali atau 1 kali untuk 1 foto untuk mendapatkan nilai informasi data tersebut
  - Informasi data yang telah di beri label disimpan pada database
- Perubahan pada data karyawan :
- Manajer melakukan ubah data nama atau jabatan karyawan
  - Manajer mengubah data yang sudah berisi informasi karyawan yang telah dicari berdasarkan ID karyawan
  - Manajer melakukan pencarian karyawan yang akan diubah atau dihapus
  - Manajer menghapus karyawan pada list karyawan

### III.5.1.2 Pencatatan Kehadiran

- Aktor : Karyawan
- Kondisi Awal : - Webcam dan sistem pencatatan kehadiran telah aktif  
- Karyawan telah terdaftar di dalam database
- Kondisi Akhir : Data kehadiran karyawan disimpan dalam database dan ditampilkan
- Skenario : - Karyawan mencapture wajahnya di depan webcam dengan kondisi frontal. Jika wajah terdeteksi bagian wajah akan di crop dengan resolusi 60x60, kemudian di *convert* ke grayscale.  
- Hasil capture di bandingkan dengan database dan mendapatkan hasil akhir yaitu karyawan telah dicatat dalam daftar hadir

### III.5.1.3 Kelola Jadwal

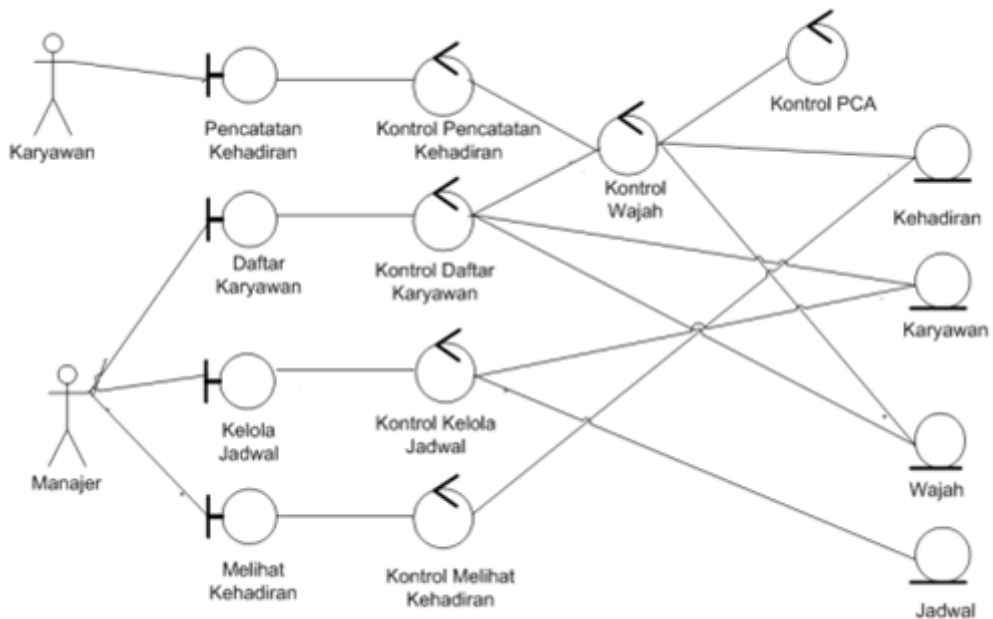
- Aktor : Manajer
- Kondisi Awal : - Sistem pencatatan kehadiran telah aktif  
- Data karyawan telah ada
- Kondisi Akhir : Jadwal karyawan sudah ada
- Skenario : - Manajer mengisi semua jadwal untuk karyawan yang telah terdaftar  
- Data jadwal karyawan yang telah diisi di simpan dalam database  
- Karyawan dapat melihat jadwal yang telah disusun oleh manajer

### III.5.1.4 Melihat Kehadiran

- Aktor : Manajer
- Kondisi Awal :  
- Sistem pencatatan kehadiran telah aktif  
- Data kehadiran telah ada
- Kondisi Akhir : Data kehadiran ditampilkan
- Skenario :  
- Manajer memilih menu melihat kehadiran karyawan  
- Manajer memilih pilihan periodic data kehadiran karyawan  
- Data kehadiran karyawan di tampilkan

### III.6 Analisis Kelas

Dalam menentukan kelas-kelas yang akan digunakan dalam aplikasi Pencatatan Kehadiran Karyawan Menggunakan Metode Wajah, analisa yang terdapat di dalamnya yaitupencatatan kehadiran, kelola karyawan, kelola jadwal dan melihat kehadiran. Berikut merupakan gambar analisis kelas :



Gambar 18 Analisis Kelas pengelolaan aplikasi

Gambar 18 menjelaskan tentang analisis kelas pada aplikasi ini. Adapun Detailan kelas – kelas pada gambar tersebut dijelaskan pada tabel4, sebagai berikut

**Tabel 4** Spesifikasi Analisis Kelas

<b>Jenis Kelas</b>	<b>Nama Kelas</b>	<b>Deskripsi</b>
Kelas Boundary	Layar Pencatatan Kehadiran	Kelas ini berhubungan dengan usecase pencatatan kehadiran
	Layar Kelola Karyawan	Kelas ini berhubungan dengan usecase kelola karyawan
	Layar Kelola Jadwal	Kelas ini berhubungan dengan usecase kelola jadwal
	Layar Melihat Kehadiran	Kelas ini berhubungan dengan usecase melihat kehadiran
Kelas Control	Pencatatan kehadiran	Mengatur akses ke dalam database
	Daftar karyawan	Mengatur akses pengelola data ke dalam database
	Kelola jadwal	Mengatur akses cetak laporan
Kelas Entity	Kehadiran	Menyimpan informasi id dan sandi
	Karyawan	Menyimpan dan mengatur informasi data karyawan
	Wajah	Menyimpan dan mengatur informasi data wajah
	Jadwal	Menyimpan dan mengatur informasi data jadwal

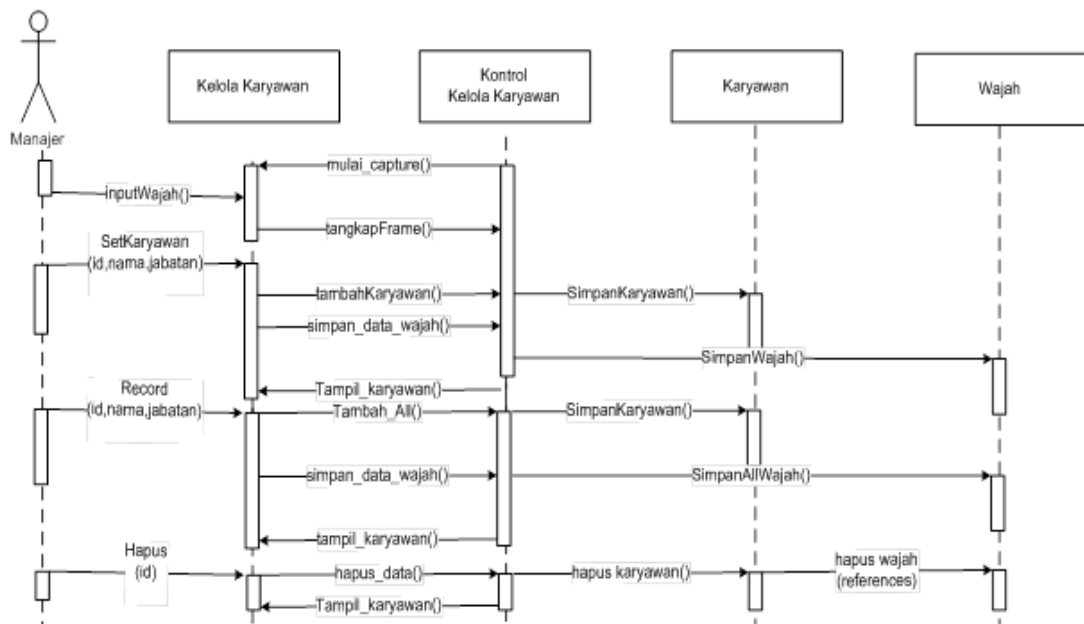
## Bab IV Perancangan

Bab ini menjelaskan mengenai *Sequence Diagram*, Kelas Diagram, Rancangan Kelas Detail, Algoritma dan Perancangan Antarmuka.

### IV.1 Sequence Diagram

Sequence diagram merupakan gambaran dari skenario use case pada bab sebelumnya yang merupakan respon dari sebuah even untuk menghasilkan output tertentu. Detail sequence diagram Sistem Pencatatan Kehadiran Karyawan dapat adalah sebagai berikut:

#### IV.1.1 Sequence Diagram Kelola Karyawan

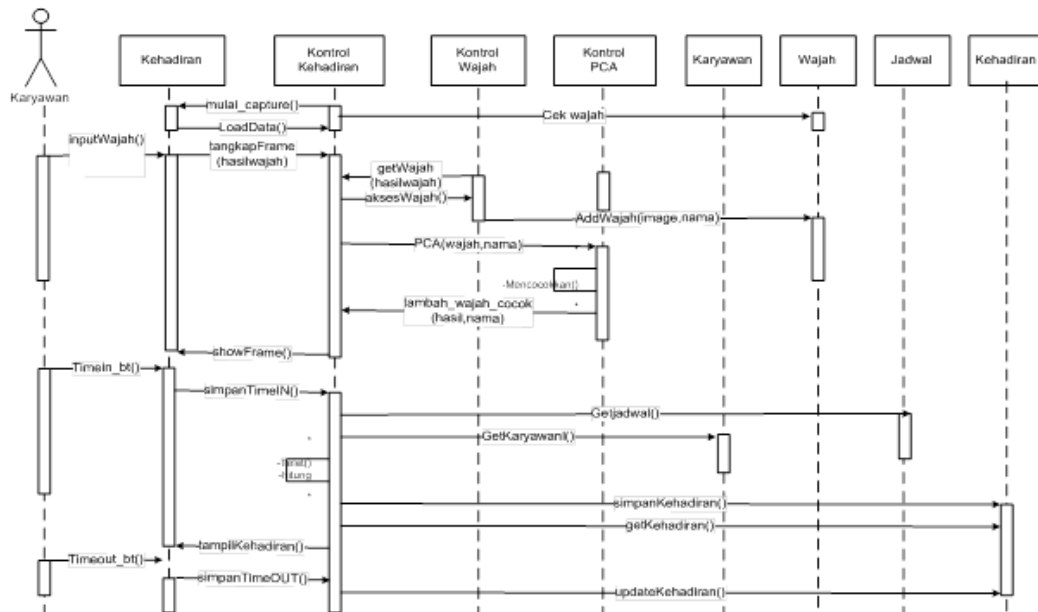


**Gambar 19 Sequence Diagram Kelola karyawan**

Pada Gambar sequence Kelola Karyawan manajer menjalankan inputwajah() dari data wajah karyawan yang dicapture, kemudian frame akan menangkap data wajah tersebut. Selanjutnya manajer memasukkan id, nama, jabatan dan menambah data karyawan. Jika manajer menjalankan Record() maka sistem merecord 10 wajah secara beruntun dan data akan ditambah 10 kali secara otomatis. Jika manajer menghapus data karyawan, manajer memasukkan id

karyawan kemudian menjalankan method `hapus_data()`. Secara otomatis data wajah yang bereferensi pada id akan terhapus.

#### IV.1.2 Sequence Diagram Kehadiran

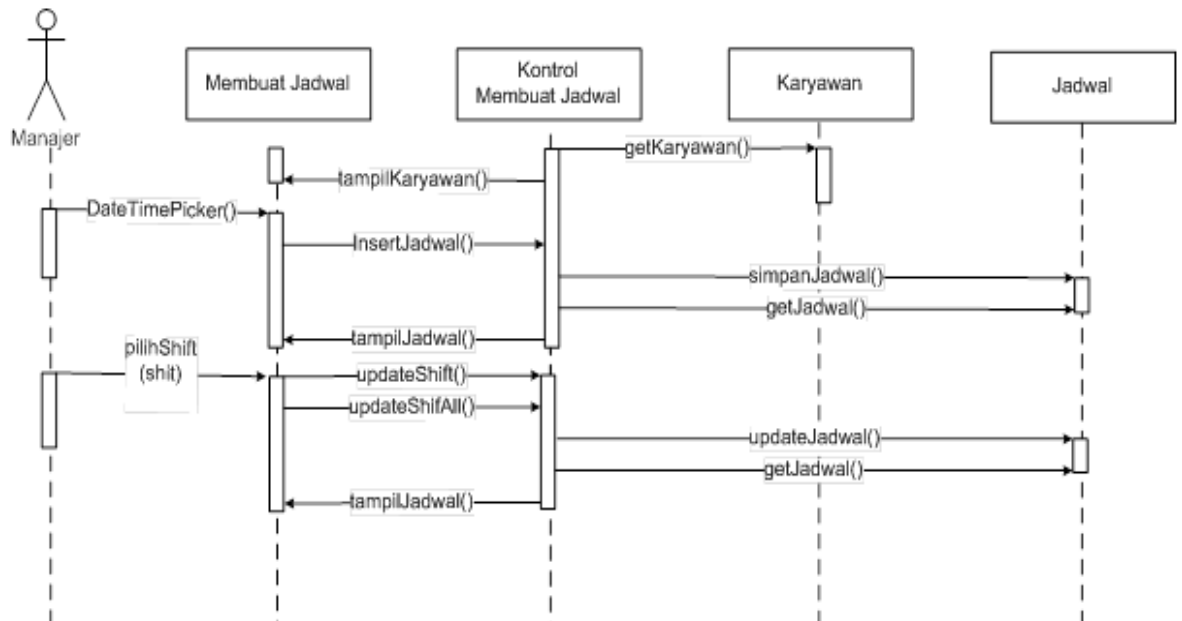


**Gambar 20 Sequence Diagram Kehadiran**

Pada gambar sequence kehadiran sistem telah menjalankan method `mulai_capture()` dan mengecek data wajah yang ada melalui `LoadData()` wajah. Sistem menangkap data wajah dari karyawan. kontrol kehadiran akan mengambil wajah pelatihan dari kontrol wajah dan menambah setiap wajah pelatihan yang telah ada melalui `aksesWajah()`. Hasil wajah akan dibandingkan melalui kontrol PCA kemudian hasilnya dikirimkan ke kontrol kehadiran untuk ditampilkan pada form kehadiran.

Jika karyawan menjalankan `timein()`, Kontrol kehadiran mencocokkan dengan data karyawan dan data shift lalu menghitung keterlambatan, kemudian menyimpannya pada database kehadiran. Jika karyawan menjalankan `timeout()`, kontrol kehadiran akan mengupdate waktu timeout karyawan kemudian menampilkan kembali pada form Kehadiran.

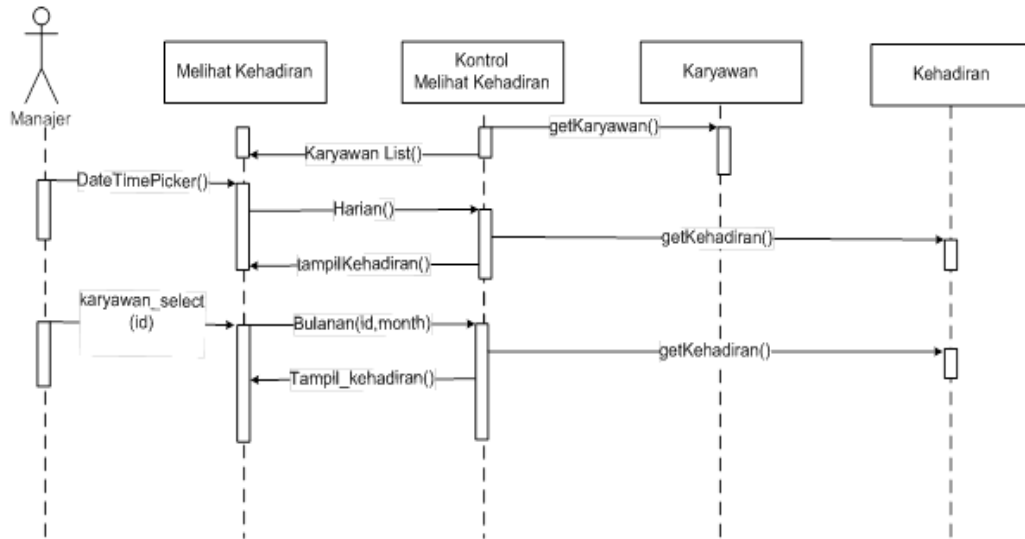
### IV.1.3 Sequence Diagram Membuat Jadwal



Gambar 21 Sequence Membuat Jadwal

Pada sequence Membuat Jadwal sistem menampilkan data karyawan yang terdaftar, kemudian manajer memilih tanggal melalui method `DateTimePicker()`. Setelah tanggal dipilih maka manajer dapat memasukkan shift atau mengupdate shift setiap karyawan dengan method `updateShift()`, atau sekaligus mengupdate seluruh shift karyawan pada tanggal yang sama dengan method `updateShiftAll()`. Kemudian sistem menampilkan hasil jadwal setiap karyawan.

#### IV.1.4 Sequence Diagram Melihat Kehadiran



**Gambar 22** Sequence Melihat Kehadiran

Pada sequence melihat kehadiran manajer dapat melihat data kehadiran karyawan yang hadir dalam hari, atau kehadiran setiap karyawan dalam sebulan. Jika manajer memilih tanggal dengan method `DateTimePicker()` maka sistem menampilkan data semua karyawan yang mencatat kehadiran pada tanggal tersebut. Jika manajer memilih `karyawan_select()` maka sistem menampilkan data kehadiran karyawan tersebut selama sebulan.

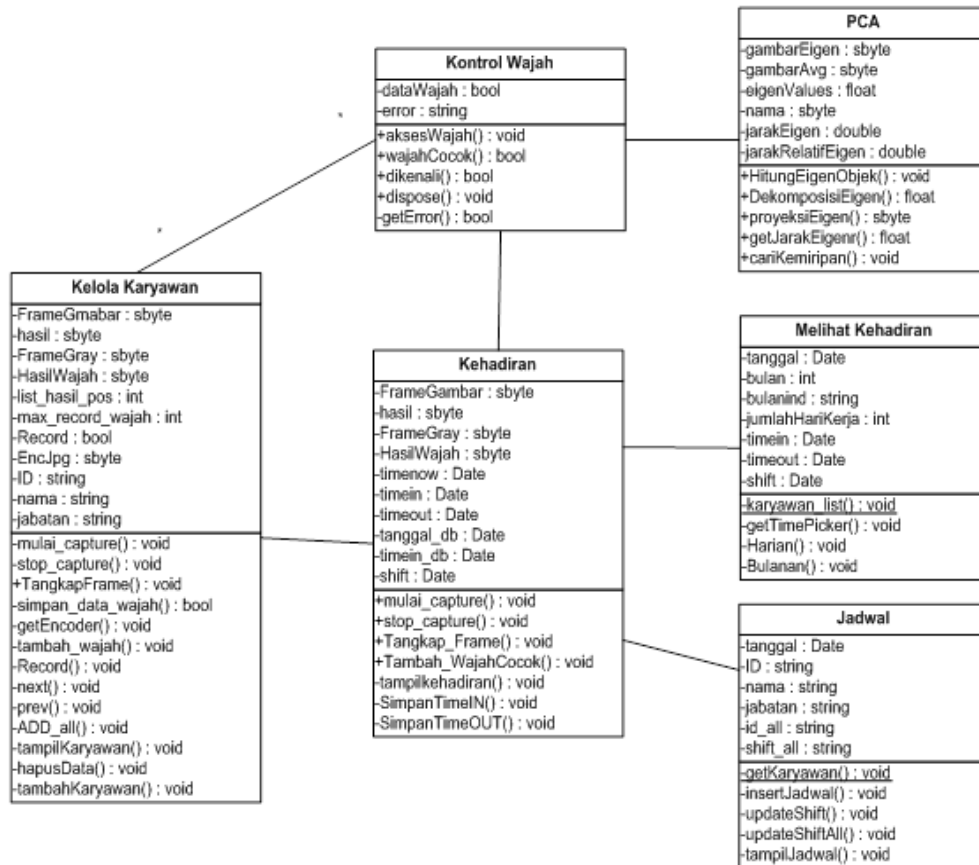
#### IV.2 Diagram Kelas

Kelas Diagram adalah diagram yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Untuk mempermudah pengelolaan hubungan antar kelas ada beberapa kelas yang digabungkan dan juga penambahan kelas kontrol. Kelas tersebut merujuk ke kelas `OpenCV`. keterangan diagram fix kelas sistem Pencatatan Kehadiran Karyawan adalah sebagai berikut:

**Tabel 5 Keterangan Kelas Fix**

<b>Kelas Fix</b>	<b>Jenis</b>	<b>Nama Kelas Sebelumnya</b>	<b>Jenis</b>	<b>Keterangan</b>
Kelola Kehadiran	Boundry	Kelola Kehadiran	Boundry	Digabungkan
		Kontrol Kelola Kehadiran	Control	
		Karyawan	Entity	
		Wajah	Entity	
Kehadiran	Boundry	Kehadiran	Boundry	Digabungkan
		Kontrol Kehadiran	Control	
		Kehadiran	Entity	
Jadwal	Boundry	Jadwal	Boundry	Digabungkan
		Kontrol Jadwal	Control	
		Jadwal	Entity	
Melihat Kehadiran	Boundry	Melihat Kehadiran	Boundry	Digabungkan
		Kontrol Melihat Kehadiran	Control	
		Kehadiran	Entity	
Kontrol Wajah		Kontrol Wajah	Control	Rujukan
Kontrol PCA		Kontrol PCA	Control	Rujukan

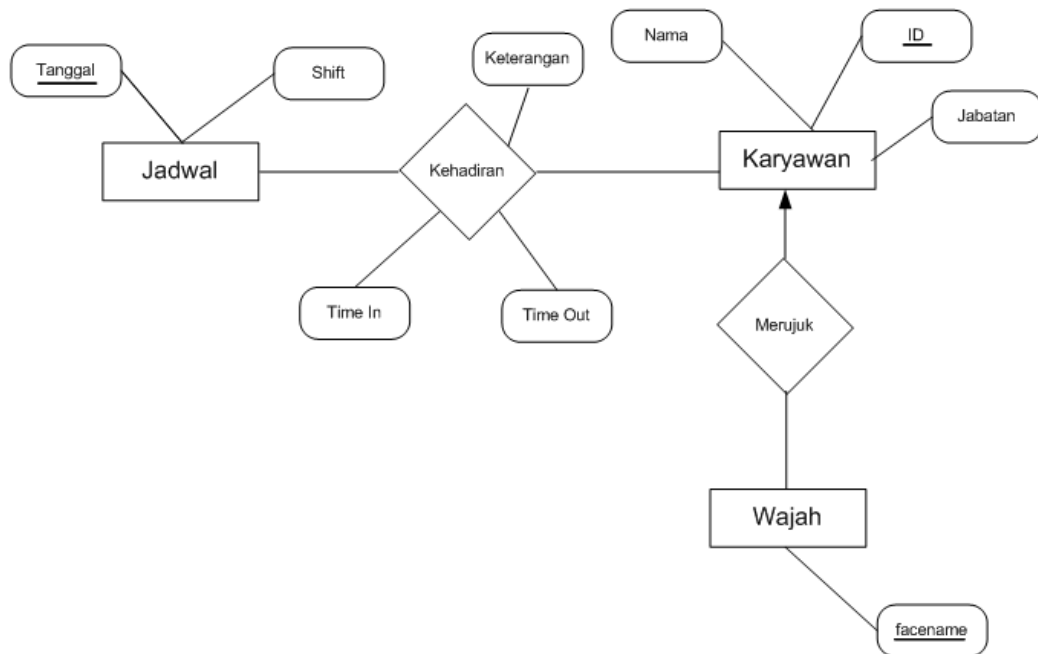
Gambar diagram kelas untuk sistem Pencatatan Kehadiran Karyawan dapat dilihat seperti pada Gambar Diagram Kelas berikut:



Gambar 23 Diagram Kelas

## IV.3 Analisis Kebutuhan Data

### IV.3.1 E-R Diagram



Gambar 24 E-R Diagram

Pada ERD diatas, terdapat 3 entitas, yaitu :

1. Entitas Karyawan memiliki 3 atribut, diantaranya ID sebagai *primary key*, Nama dan Jabatan. Terhubung dengan entitas Jadwal dan Wajah. Kardinalitas dengan entitas Jadwal adalah *many to many*. Kardinalitas dengan entitas Wajah adalah *many to many*.
2. Entitas Jadwal memiliki 2 atribut, diantaranya Tanggal sebagai *primary key* dan shift.
3. Entitas Wajah memiliki 1 atribut, yaitu facename sebagai *primary key*. Entitas ini merupakan entitas lemah karena *primary key* yang ada di entitas ini bergantung pada entitas yang lain, yaitu entitas Karyawan.

## IV.4 Rancangan Kelas Detail

### IV.4.1 Kelas Kelola Karyawan

#### IV.4.1.1 Detail Kelas Karyawan

Nama operasi : Kelola Karyawan

Kelola Karyawan
-FrameGmabar : sbyte -hasil : sbyte -FrameGray : sbyte -HasilWajah : sbyte -list_hasil_pos : int -max_record_wajah : int -Record : bool -EncJpg : sbyte -ID : string -nama : string -jabatan : string
-mulai_capture() : void -stop_capture() : void +TangkapFrame() : void -simpan_data_wajah() : bool -getEncoder() : void -tambah_wajah() : void -Record() : void -next() : void -prev() : void -ADD_all() : void -tampilKaryawan() : void -hapusData() : void -tambahKaryawan() : void

#### IV.4.1.2 Algoritma Kelas Kelola Karyawan

Algoritma : Kelola Karyawan

<i>*Deklarasi*</i>
<i>Var</i>
<i>frameGambar,frameGray :image</i> <i>hasilwajah,EncJpg : image&lt;list&gt;</i> <i>ID,nama,jabatan,faceName : string</i> <i>Record : bool</i>
<i>*Deskripsi*::Method TangkapFrame()</i>
<i>read (wajah)//menggunakan face detection</i> <i>TangkapFrame(frameGambar)</i>

<pre> if (framegambar != null)THEN     frameGray = frameGambar.convert&lt;gray&gt;     foreach (frameGray in hasil)         hasilWajah = hasil     endif </pre>
<pre> *Deskripsi*::Method Record()  //capture Begin Read (ID,nama,jabatan) if (hasilWajah.Count = 10)     for (i=0;i&lt;hasilWajah.count;i++)         simpan_data_wajah[i]         write ("tambah wajah karyawan ke-",i)     endif End </pre>
<pre> *Deskripsi*::Method TambahKaryawan()  ID = ID_txt nama = nama_txt jabatan = jabatan_cb //koneksi database Begin INSERT INTO karyawan VALUES(ID,nama, jabatan) Write ("Database telah masuk ke karyawan") End </pre>
<pre> *Deskripsi*::simpan data wajah()  nama = nama_txt ID = ID_txt //koneksi database Begin Create facename = "face"+nama+random+".jpg" INSERT INTO wajah VALUES( facename ,ID ,hasilWajah) SELECT count(facename) FROM wajah where id= ID max = getValue (facename) write ("Database wajah ke-,max,ditambah") End </pre>
<pre> *Deskripsi*::Method hapusData()ID  Read (ID_txt) ID = ID_txt; //koneksi database Begin DELETE FROM karyawan WHERE id=ID tampilKaryawan End </pre>
<pre> *Deskripsi*::Method tampilKaryawan() </pre>

```

Create tabel (colom ID, colom nama,colom jabatan)
Begin
//koneksi database
SELECT id,nama,jabatan FROM karyawan
int = 0
While (dataditemukan)
  Write (Colom id.r0w[i] = id)
  Write (Colom nama.r0w[i] = nama)
  Write (Colom jabatan.r0w[i] = jabatan)
i++

```

#### IV.4.2 Kelas Kehadiran

##### IV.4.2.1 Detail Kelas Kehadiran

Nama operasi : Kehadiran

Kehadiran
-FrameGambar : sbyte -hasil : sbyte -FrameGray : sbyte -HasilWajah : sbyte -timenow : Date -timein : Date -timeout : Date -tanggal_db : Date -timein_db : Date -shift : Date
+mulai_capture() : void +stop_capture() : void +Tangkap_Frame() : void +Tambah_WajahCocok() : void -tampilkehadiran() : void -SimpanTimeIN() : void -SimpanTimeOUT() : void

##### IV.4.2.2 Algoritma Kelas Kehadiran

Algoritma : Kehadiran

```

*Deklarasi*
Var
frameGambar,frameGray,hasil:image
ID,nama,jabattan,facename : string
Timenow,timein,timeout : datetime
Timein_db,shift : timespan

```

*\*Deskripsi\*::Method loadData ()*

*Eigenface = kontrol\_wajah()*

*if (eigenface.wajahcocok= true) THEN*

*write ("data karyawan telah ada")*

*endif*

*else*

*write ("data karyawan kosong, pilih kelola karyawan")*

*\*Deskripsi\*::Method TangkapFrame()*

*read (wajah) //menggunakan face detection*

*frameGambar = wajah*

*TangkapFrame(frameGambar)*

*if (framegambar != null)THEN*

*frameGray = frameGambar.convert<gray>*

*foreach (frameGray in hasil)*

*hasilWajah = hasil*

*if (tangkapFrame=hasil) THEN*

*write frameGambar(nama)*

*tambah\_wajah\_cocok (image,nama)*

*endif*

*\*Deskripsi\*::Method SimpanTimeIN()*

*timenow = DateTime.Now;*

*timein = timenow;*

*begin*

*frameGambar(nama)*

*nama\_txt = nama*

*//koneksi database*

*SELECT \* FROM jadwal where nama = nama\_text and tanggal = tanggal\_now*

*if (data sudah ada)*

*if ((tanggaldb = tanggalnow)) THEN*

*if (nama= ID2 ) THEN*

*Write ("Anda Sudah Absen Masuk Hari ini")*

*endif*

*if (nama\_text != ID2 or ID2 = null) THEN*

*int telatjam = jamnow-jamdb;*

*int telatmin = minnow - mindb;*

*else if ((telatjam <= 0) && (telatmin<=0))*

*keterangan = "Tepat Waktu"*

*else if ((telatjam < 0) && (telatmin > 0))*

*keterangan = "Tepat Waktu"*

*else*

*keterangan = "Telat ",telatjam+telatmenit*

*INSERT INTO kehadiran VALUES (ID,nama,jabatan,timein,null,keterangan)*

*Write ("Terima kasih Anda Telah Absen Masuk jam, timenow)*

*endif*

*else*

```

INSERT INTO kehadiran VALUES (ID,nama,jabatan,timein,null,keterangan)
Write ("Terima kasih Anda Telah Absen Masuk jam, timenow)
endif
END

*Deskripsi*::Method SimpanTimeOUT()
timenow = DateTime.Now;
timeout = timenow;
begin
frameGambar(nama)
//koneksi database
update kehadiran set timeout =timenow where nama = nama_text
if (timein_db != null) THEN
    Write ("Terima Kasih , nama_text, Anda Telah Absen Pulang jam: timenow")
endif
else
    Write("Anda Belum Absen Masuk");

```

### IV.4.3 Kelas Jadwal

#### IV.4.3.1 Detail Kelas Jadwal

Nama operasi : membuat jadwal

Jadwal
-tanggal : Date
-ID : string
-nama : string
-jabatan : string
-id_all : string
-shift_all : string
-getKaryawan() : void
-insertJadwal() : void
-updateShift() : void
-updateShiftAll() : void
-tampilJadwal() : void

#### IV.4.3.2 Algoritma Kelas Jadwal

Algoritma : Jadwal

<pre><i>*Deskripsi*::Method getKaryawan ()</i> // koneksi database "SELECT ID,nama,jabatan FROM karyawan While (data ada)   ID = get(ID)   nama = get(nama)   jabatan = get(jabatan)   shift = 'noshift' tampilJadwal</pre>
<pre><i>*Deskripsi*::Method insertJadwal()</i> // koneksi database SELECT ID FROM jadwal where id = id_text if (ID == null) THEN   INSERT INTO jadwal VALUES(ID,nama,jabatan,shift) endif else write ("jadwal tanggal ini sudah dibuat") tampilJadwal</pre>
<pre><i>*Deskripsi*::Method UpdateShift()</i> tanggal = dateTimePicker id = column_click // koneksi database update jadwal set shift = shift_cb where ID = id and tanggal = tanggal write ("shift disimpan")</pre>
<pre><i>*Deskripsi*::Method UpdateShiftAll()</i> tanggal = dateTimePicker // koneksi database Begin SELECT count(ID) FROM karyawan";   int max = 0;   while (data ada)     max = get(ID) SELECT * FROM jadwal   while (i &lt; max)     id_all [max] = get (id)     shift_all[max] = get(shift) update jadwal set shift = shift_all where ID = id_all and tanggal = tanggal   i++ write ("Jadwal Tanggal ini Disimpan") end</pre>

#### IV.4.4 Kelas Melihat Kehadiran

##### IV.4.4.1 Detail Kelas Melihat Kehadiran

Nama operasi : Melihat Kehadiran

Melihat Kehadiran
-tanggal : Date
-bulan : int
-bulanind : string
-jumlahHariKerja : int
-timein : Date
-timeout : Date
-shift : Date
-karyawan_list() : void
-getTimePicker() : void
-Harian() : void
-Bulanan() : void

##### IV.4.4.2 Algoritma Kelas Melihat Kehadiran

Algoritma : Melihat Kehadiran

<i>*Deklarasi*</i> <i>Var</i> <i>tanggal : datetime</i> <i>bulan, jumlahHariKerja : int</i> <i>bulanind : string</i> <i>timein, timeout, shift : timespan</i> <i>Parent Kehadiran</i>
<i>*Deskripsi*::Method Harian()</i> <i>Tanggal = dateTimePicker</i> <i>Create tabelJadwal()</i> <i>// koneksi database</i>  <i>SELECT * FROM kehadiran where tanggal = Tanggal</i> <i>while (data ada)</i> <i>tabelJadwal.Rows[i].Cells[id]= get(id)</i> <i>tabelJadwal.Rows[i].Cells[nama]= get(nama)</i> <i>tabelJadwal.Rows[i].Cells[jabatan]= get(jabatan)</i> <i>tabelJadwal.Rows[i].Cells[tanggal]= get (tanggal)</i> <i>tabelJadwal.Rows[i].Cells[hari] = get(hari)</i> <i>tabelJadwal.Rows[i].Cells[shift]= get(shift)</i> <i>tabelJadwal.Rows[i].Cells[timein]= get(timein)</i> <i>tabelJadwal.Rows[i].Cells[timeout]= get(timeout)</i> <i>tabelJadwal.Rows[i].Cells[keterangan] = get(keterangan)</i> <i>i = i + 1</i>

\*Deskripsi\*::Method getTimePicker()

*tanggal = dateTimePicker.date;*

*bulan = dateTimePicker.Month;*

*Begin*

*switch month*

*case 1:*

*write ("januari")*

*case 2 :*

*write ("februari")*

*...*

*case 12 :*

*write ("Desember")*

*Default :*

*Write ("pilih bulan")*

*Harian(tanggal)*

*End*

\*Deskripsi\*::Method karyawanList()

*Bulan = dateTimePicker.month*

*//koneksi database*

*int i = 0;*

*SELECT ID,nama FROM karyawan";*

*while (data ada)*

*listBox.Items.Add(ID+nama);*

*i = i + 1;*

*if (listbox.item.selected)*

*{*

*id = listBox\_cb*

*Bulanan(id,bulan);*

*}*

*Endif*

\*Deskripsi\*::Method Bulanan(id,month)

*Tanggal = dateTimePicker*

*Create tabelJadwal()*

*SELECT \* FROM kehadiran where ID = 'id' and tanggal like '%-+month+"-%'*

*while (data ada)*

*tabelJadwal.Rows[i].Cells[id]= get(id)*

*tabelJadwal.Rows[i].Cells[nama]= get(nama)*

*tabelJadwal.Rows[i].Cells[jabatan]= get(jabatan)*

*tabelJadwal.Rows[i].Cells[tanggal]= get (tanggal)*

*tabelJadwal.Rows[i].Cells[hari] = get(hari)*

*tabelJadwal.Rows[i].Cells[shift]= get(shift)*

*tabelJadwal.Rows[i].Cells[timein]= get(timein)*

*tabelJadwal.Rows[i].Cells[timeout]= get(timeout)*

*tabelJadwal.Rows[i].Cells[keterangan] = get(keterangan)*

*i = i + 1*

## IV.4.5 Kelas Kontrol Wajah

### IV.4.5.1 Detail Kelas Kontrol Wajah

Nama operasi : Kontrol Wajah

Kontrol Wajah
-dataWajah : bool
-error : string
+aksesWajah() : void
+wajahCocok() : bool
+dikenali() : bool
+dispose() : void
-getError() : bool

### IV.4.5.2 Algoritma Kelas Kontrol Wajah

Algoritma : Kontrol Wajah

<i>*Deklarasi*</i> <i>Var</i> <i>dataWajah : bool</i> <i>error : string</i>
<i>*Deskripsi*::Method AksesWajah(loKasi)</i> <i>//koneksi database</i> <i>if (lokasi wajah ada)</i> <i>  select * from wajah</i> <i>  face = get(image)</i> <i>  while (data ada)</i> <i>    face = get(facename)</i> <i>    List_Nama.Add(get(nama));</i> <i>    ImagePelatihan.Add(face)</i> <i>    i++</i> <i>if (ImagePelatihan not null) THEN</i> <i>  Pengenalan = new PCA(ImagePelatihan,</i> <i>    List_Nama)</i> <i>Endif</i>

## IV.4.6 Kelas PCA

### IV.4.6.1 Detail Kelas PCA

Nama operasi : PCA

PCA
-gambarEigen : sbyte
-gambarAvg : sbyte
-eigenValues : float
-nama : sbyte
-jarakEigen : double
-jarakRelatifEigen : double
+HitungEigenObjek() : void
+DekomposisiEigen() : float
+proyeksiEigen() : sbyte
+getJarakEigenr() : float
+cariKemiripan() : void

### IV.4.6.2 Algoritma Kelas PCA

Algoritma : PCA

<i>*Deklarasi*</i> Var gambarEigen, gambarRata2 : images[] eigenValues : float[] nama : string JarakEigen : double
<i>*Deskripsi*::Method HitungEigenObjek(ListGambar)</i> //mengambil resolusi setiap gambar pelatihan untuk menentukan maksimum iterasi //membuat objek untuk menampung semua gambar pelatihan dalam bentuk Array //membuat instansiasi gambar Eigen dalam bentuk Array //membuat variabel avg untuk menampung gambar rata-rata
<i>*Deskripsi*::Method</i> <i>DekomposisiEigen(gambarTest, gambarEigen, gambarRata2)</i> //mendekomposisi setiap gambar sebagai nilai eigenvalues dari eigenvectors //return eigenvalues
<i>*Deskripsi*::Method GetJarakEigen(gambarTest)</i> //memanggil method Dekomposisi Eigen <i>DekomposisiEigen(gambarTest, gambarEigen, gambarRata2)</i> //hitung jarak Eigen antara parameter gambarTest dengan gambar lain dalam gambar pelatihan didatabase // return Array jarak Eigen dari setiap gambar
<i>*Deskripsi*::Method CariKemiripan(gambarTes, nama, jarakEigen)</i> //memanggil method GetJarakEigen

```

//hasil nilai jarak Eigen disimpan pada Array Jarak
Jarak []= GetJarakEigen(gambarTests);
index = 0;
//mencari nilai minimum dengan acuan nilai gambar array[0]
//jika ada yang nilai yang lebih kecil maka looping
jarakEigen = Jarak[0];
  for (int i = 1; i < Jarak.Length; i++)
    if (Jarak[i] < jarakEigen) THEN
      jarakEigen = Jarak[i]
//set nilai terkecil dari indeks pada variable index
  index = i;
  Endif

```

---

```

*Deskripsi*::Method Mencocokkan (gambar)
//mengambil gambar test dari parameter
gambarTest = gambar
jarakEigen = 0
//memanggil method CariKemiripan dengan mengirim parameter gambarTest
// return nama[index]

```

## IV.5 Perancangan Antarmuka

### IV.5.1 Kelola Karyawan

#### IV.5.1.1 Rancangan Antarmuka



Gambar 25 Antarmuka Kelola Karyawan

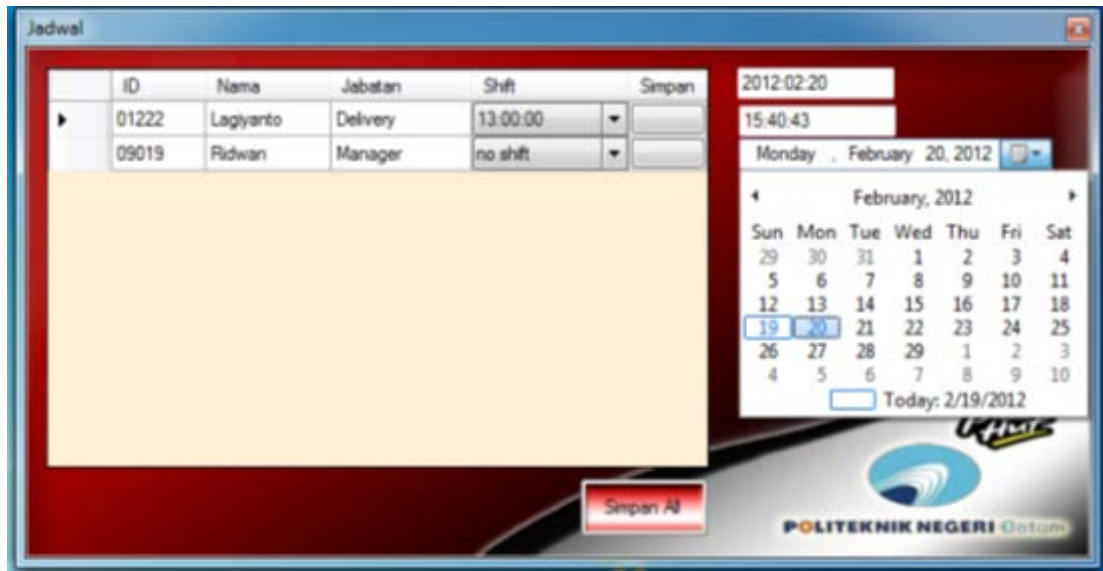
#### IV.5.1.2 Deskripsi Rancangan

Tabel 6 Deskripsi Rancangan Antarmuka Kelola Karyawan

Id_Objek	Jenis	Nama	Keterangan
Frame_Box	Picture Box	Wajah Box	Menampilkan wajah yang dicapture
Wajah_Box	Picture Box	Wajah Box	Menampilkan wajah hasil capture
Tambah_wajah	Button	Tambah Wajah	Menambah gambar yang di simpan ke database
Capture_BTN	Button	Capture	Mengambil wajah yang telah di capture dengan camera
Button1	Button	Refresh	Mengupdate data
Button2	Button	Delete Folder	Menghapus hasil capture di folder
ID_txt	TextBox	ID	Menampilkan id setiap karyawan
Nama_txt	TextBox	Nama	Menampilkan Nama karyawan
Jabatan_cb	TextBox	Jabatan	Menampilkan jabatan karyawan
DataGrideView 1	DataGridView	Tabel data karyawan	Menampilkan id, nama, jabatan, shift setiap karyawan
Button3	Button	Refresh	Menampilkan data karyawan

## IV.5.2 Jadwal

### IV.5.2.1 Rancangan Antarmuka



Gambar 26 Antarmuka Jadwal

### IV.5.2.2 Deskripsi Rancangan

Tabel 7 Deskripsi Rancangan Antarmuka Jadwal

Id_Objek	Jenis	Nama	Keterangan
<i>Button1</i>	<i>Button</i>	<i>SimpanAll</i>	<i>Menyimpan jadwal yang telah dibuat ke database</i>
<i>DateTimePicker1</i>	<i>DateTimePicker</i>	<i>Tanggal</i>	<i>Menampilkan tanggal sekarang</i>
<i>TextBox1</i>	<i>TextBox</i>	<i>Jam masuk</i>	<i>Menampilkan jam masuk</i>
<i>TextBox2</i>	<i>TextBox</i>	<i>Jam keluar</i>	<i>Menampilkan jam keluar</i>
<i>DataGriev1</i>	<i>DataGridView</i>	<i>Tabel data karyawan</i>	<i>Menampilkan id, nama, jabatan, shift setiap karyawan</i>

### IV.5.3 Kehadiran

#### IV.5.3.1 Rancangan Antarmuka



Gambar 27 Antarmuka Kehadiran

#### IV.5.3.2 Deskripsi Kehadiran

Tabel 8 Deskripsi Rancangan Antarmuka Kehadiran

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>Image_PICBX</i>	<i>Picture Box</i>	<i>Wajah Box</i>	<i>Menampilkan wajah yang dicapture</i>
<i>Box_Wajahcocok</i>	<i>Picture Box</i>	<i>Wajah Box</i>	<i>Menampilkan wajah hasil capture</i>
<i>Timein_bt</i>	<i>Button</i>	<i>TimeIN</i>	<i>Memasukkan jam masuk</i>
<i>Timeout_bt</i>	<i>Button</i>	<i>TimeOut</i>	<i>Memasukkan jam keluar</i>
<i>Time_box</i>	<i>TextBox</i>	<i>Jam sekarang</i>	<i>Menampilkan waktu sekarang</i>
<i>Day_box</i>	<i>TextBox</i>	<i>Hari sekarang</i>	<i>Menampilkan nama hari</i>
<i>DataGridView1</i>	<i>DataGridView</i>	<i>Tabeldatakaryawan</i>	<i>Menampilkan id, nama, jabatan, shift, timeIN, TimeOut, Keterangan setiap karyawan</i>
<i>Loading</i>	<i>loading</i>	<i>DataGridView</i>	<i>Menampilkan Loading</i>

## IV.5.4 Melihat Kehadiran

### IV.5.4.1 Rancangan Antarmuka



Gambar 28 Antarmuka Melihat Kehadiran

### IV.5.4.2 Deskripsi Kehadiran

Tabel 9 Deskripsi Rancangan Antarmuka Melihat Kehadiran

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
DateTimePicker 1	<i>DateTimePicker</i>	<i>Tanggal</i>	<i>Menampilkan tanggal sekarang</i>
<i>textBox1</i>	<i>TextBox</i>	<i>Date</i>	<i>Menampilkan nama karyawan</i>
<i>textBox2</i>	<i>TextBox</i>	<i>name</i>	<i>Menampilkan jabatan karyawan</i>
<i>ListBox1</i>	<i>TextBox</i>	<i>ListBox</i>	<i>Menampilkan nama-nama setiap karyawan</i>
<i>textBox3</i>	<i>TextBox</i>	<i>Jumlah hari kerja</i>	<i>Menampilkan jumlah kerja</i>
DataGridView1	<i>DataGridView</i>	<i>Tabel data karyawan</i>	<i>Menampilkan id, nama, jabatan, tanggal, hari, shift, Timein, timeOut, keterangan setiap karyawan</i>

## 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. Adapun fungsi yang akan diuji pada sistem Pencatatan Kehadiran Karyawan sesuai dengan analisis kebutuhan fungsional adalah sebagai berikut:

1. Sistem menerima masukan data gambar wajah dari karyawan
2. Sistem mampu mencari dan mengolongkan data yang memiliki kemiripan
3. Sistem menyediakan fitur, menambah dan menghapus karyawan
4. Sistem menyediakan fitur pengelolaan jadwal karyawan
5. Sistem menampilkan informasi pencatatan kehadiran karyawan

### V.1 Implementasi Kelas

Tabel 10 Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	Kelas Kelola Karyawan	Kelola Karyawan.cs	Kelola Karyawan.dll
2	Kelas Kehadiran	Kehadiran.cs	Kehadiran.dll
3	Kelas Jadwal	Jadwal.cs	Jadwal.dll
4	Kelas Melihat Kehadiran	Melihat Kehadiran.cs	Melihat Kehadiran.dll
5	Kelas Kontrol Wajah	Kontrol Wajah.cs	Kontrol Wajah.dll
6	Kelas PCA	PCA.cs	PCA.dll

## V.2 Pengujian Pengenalan Wajah

### V.2.1 Skenario Pengujian Pengenalan wajah

Kelas yang diuji:

Nama kelas :

- Kelas Kehadiran
- Kelas PCA
- Kelas Kontrol Wajah

Pada tahapan yang pertama skenario pengenalan wajah

1. Mempersiapkan sebuah objek berupa wajah manusia. Objek manusia yang akan diambil terdiri dari 6 sampel data wajah manusia yang terdiri dari 2 wajah pria dan 4 wajah wanita.
2. Setiap orang memiliki 10 database wajah pelatihan yang diambil secara beruntun dan setiap orang akan diambil data uji sebanyak 10 kali
3. Objek akan diuji pada 2 kondisi penerangan terang dan pencahayaan gelap

### V.2.2 Hasil Pengujian Pengenalan wajah

Hasil pengujian identifikasi wajah pencahayaan terang dapat dilihat pada tabel Berikut:

**Tabel 11 pengamatan I**

Kondisi	Terang									
	Non-Ekspresi									
Pengamatan	1	2	3	4	5	6	7	8	9	10
Ridwan	√	√	√	√	X	√	√	√	√	√
Lagiyanto	√	√	√	√	√	√	√	√	√	√
Kurnia	√	√	√	√	√	√	X	√	√	√
Sarmi	√	√	√	√	√	√	√	√	√	√
Putri	√	√	√	√	√	√	√	√	√	√
Tini	√	√	√	√	√	√	√	√	√	√

**Tabel 12 pengamatan II**

Kondisi	Terang Ekspresi									
	1	2	3	4	5	6	7	8	9	10
Pengamatan	1	2	3	4	5	6	7	8	9	10
Ridwan	√	√	√	√	X	√	√	√	√	√
Lagiyanto	√	√	√	√	√	√	X	√	√	√
Kurnia	√	√	√	√	√	√	√	√	√	√
Sarmi	√	√	√	√	√	√	√	X	√	√
Putri	√	√	√	√	√	√	√	√	√	√
Tini	√	√	√	√	√	√	√	√	√	√

**Tabel 13 pengamatan III**

Kondisi	Terang Dengan Accesories (Kacamata + Topi)									
	1	2	3	4	5	6	7	8	9	10
Pengamatan	1	2	3	4	5	6	7	8	9	10
Ridwan	√	√	√	√	X	√	√	√	X	√
Lagiyanto	√	√	√	√	√	√	X	√	√	X
Kurnia	√	√	√	√	√	√	√	√	X	-
Sarmi	√	√	√	√	X	√	√	√	-	√
Putri	√	√	√	√	√	√	√	√	√	√
Tini	√	√	√	√	√	√	√	√	√	X

Keterangan:

√ = wajah teridentifikasi

X = wajah yang teridentifikasi tidak sesuai

- = wajah tidak teridentifikasi

## Analisa

1. Dari percobaan 1 diatas, ditemukan kegagalan sebanyak 2 kali dari sebanyak 60 kali percobaan. Percobaan tanpa ekspresi, wajah teridentifikasi dengan baik. Sehingga dapat di hitung

$$60 - 2 = 48 \times 2 : 180 = 96\%$$

2. Percobaan ke 2 dengan ekspresi, wajah teridentifikasi tidak sempurna karena terdapat 3 kegagalan dari 60 kali percobaan,wajah tidak teridentifikasi dengan baik. Dapat di hitung

$$60 - 3 = 47 \times 2 : 180 = 94\%$$

3. Percobaan ke 3 dengan menggunakan aksesoris kaca mata, topi, dan jilbab, wajah teridentifikasi tidak sempurna, karena terdapat 7 kesalahan dan 2 tidak teridentifikasi sama sekali.

$$60 - 9 = 41 \times 2 : 180 = 81\%$$

Data pengamatan di atas didapat berdasarkan hasil percobaan dengan *background* yang sama yaitu, pencahayaan yang terang, dilakukan pada siang hari dan wajah yang di-*capture* dengan 3 tipe yaitu tanpa ekspresi, dan dengan ekspresi serta dengan aksesoris. Percobaan dilakukan pada 6 orang, setiap orang memiliki *database* wajah sebanyak 10 wajah dan dilakukan percobaan sebanyak 6 kali setiap orang. Berikut ini hasil capture yang dilakukan, terdapat pada gambar 29 hasil pengujian pada siang hari

**Gambar 29** hasil pengujian pada siang hari





Hasil pengujian identifikasi wajah pencahayaan gelap dapat dilihat pada tabel Berikut:

**Tabel 14 pengamatan IV**

Kondisi	Gelap Non-Ekspresi									
	1	2	3	4	5	6	7	8	9	10
Pengamatan	1	2	3	4	5	6	7	8	9	10
Ridwan	√	√	√	√	X	√	√	√	√	√
Lagiyanto	√	√	√	√	√	√	√	√	√	√
Kurnia	√	√	√	√	√	√	X	√	√	√
Sarmi	√	√	√	√	√	√	√	X	√	√
Putri	√	√	√	√	√	√	√	√	√	√
Tini	√	√	√	√	√	√	√	X	√	√

**Tabel 15 pengamatan V**

Kondisi	Gelap Ekspresi									
	1	2	3	4	5	6	7	8	9	10
Pengamatan	1	2	3	4	5	6	7	8	9	10
Ridwan	√	√	√	√	X	√	√	√	√	√
Lagiyanto	√	√	√	√	√	√	√	√	X	√
Kurnia	√	√	√	√	√	√	X	√	√	√
Sarmi	√	√	X	√	√	√	√	X	√	√
Putri	√	√	√	√	√	√	√	√	√	√
Tini	√	√	√	√	√	√	√	X	√	√

**Tabel 16 pengamatan VI**

Kondisi	Gelap Dengan Accesories (Kacamata + Topi)									
	1	2	3	4	5	6	7	8	9	10
Pengamatan	1	2	3	4	5	6	7	8	9	10
Ridwan	√	√	√	√	X	√	√	√	-	√
Lagiyanto	√	√	√	√	√	√	√	-	√	X
Kurnia	√	√	√	-	√	√	X	√	√	√
Sarmi	√	√	X	√	√	X	√	√	-	√
Putri	√	√	√	√	X	√	√	√	√	√
Tini	√	√	√	√	√	X	√	X	√	√

Keterangan:

√ = wajah teridentifikasi

X = wajah yang teridentifikasi tidak sesuai

- = wajah tidak teridentifikasi

## Analisa

1. Dari percobaan 1 diatas, ditemukan kegagalan sebanyak 4 kali dari sebanyak 60 kali percobaan. Percobaan tanpa ekspresi, wajah teridentifikasi dengan baik. Sehingga dapat di hitung

$$60 - 4 = 56 \times 2 = 92\%\%$$

2. Percobaan ke 2 dengan ekspresi, wajah teridentifikasi tidak sempurna karena terdapat 6 kegagalan dari 60 kali percobaan,wajah tidak teridentifikasi dengan baik. Dapat di hitung

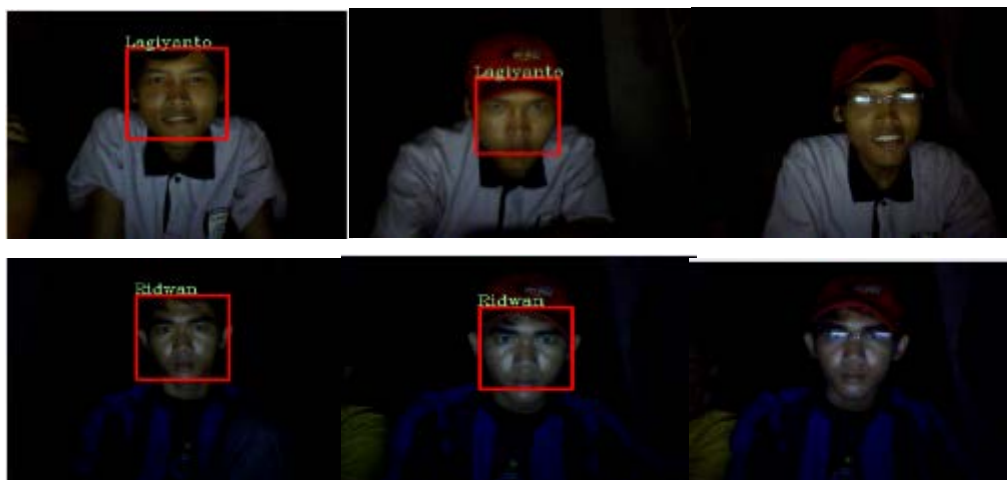
$$60 - 3 = 47 \times 2 : 180 = 90\%$$

3. Percobaan ke 3 dengan menggunakan aksesoris kaca mata, topi, dan jilbab, wajah teridentifikasi tidak sempurna, karena terdapat 12 kesalahan dan 3 tidak teridentifikasi sama sekali.

$$60 - 11 = 39 \times 2 : 180 = 80\%$$

Data pengamatan didapat berdasarkan hasil percobaan dengan *background* yang sama yaitu , pencahayaan yang kurang, di lakukan pada malam hari dan wajah yang di-*capture* dengan 3 tipe yaitu tanpa ekspresi, dan dengan ekspresi sertadengan accesories. Percobaan dilakukan pada 3 orang, setiap orang memiliki *database* wajah sebanyak 10 wajah dan dilakukan percobaan sebanyak 6 kali setiap orang. Berikut ini hasil capture yang di lakukan, terdapat pada gambar 30

Gambar 30 Hasil pengujian pada malam hari





Sehingga hasil dari akumulasi dari semua kondisi hasil pengujian adalah rata-rata dari hasil sub pengujian yaitu:

1. Pengujian dengan kondisi pencahayaan terang

$$\text{Acc} = (\text{Pengujian I}) + (\text{Pengujian II}) + (\text{Pengujian III}) / \text{Jumlah pengujian}$$

$$\text{Acc} = 96\% + 94\% + 81\% / 3$$

$$\text{Acc} = 90,3\%$$

2. Pengujian dengan kondisi pencahayaan gelap

$$\text{Acc} = (\text{Pengujian IV}) + (\text{Pengujian V}) + (\text{Pengujian VI}) / \text{Jumlah Pengujian}$$

$$\text{Acc} = 92\% + 90\% + 80\% / 3$$

$$\text{Acc} = 87,3\%$$

3. Pengujian dengan kondisi pencahayaan terang dan gelap

$$\text{Acc} = (\text{Pengujian I}) + (\text{Pengujian II}) + (\text{Pengujian III}) + (\text{Pengujian IV}) + (\text{Pengujian V}) + (\text{Pengujian VI}) / \text{Jumlah pengujian}$$

$$\text{Acc} = 96\% + 94\% + 81\% + 92\% + 90\% + 80\% / 6$$

$$\text{Acc} = 88\%$$

## V.3 Pengujian Kelola Karyawan

### V.3.1 Skenario Pengujian Kelola Karyawan

Kelas yang diuji:

Nama kelas :

- Kelas Kelola Karyawan
- Kelas Kontrol Wajah

Skenario Kelola karyawan adalah:

1. Manajer melakukan registrasi dengan password.
2. Manajer memilih 3 menu yaitu kelola karyawan, jadwal dan laporan kehadiran.

3. Ketika manajer memilih menu kelola Karyawan, maka manajer mendaftarkan data karyawan baru kedalam database. Data-data yang dimasukkan kedalam database terdiri dari: hasil capture wajah, ID, Nama dan Jabatan.
4. Ketika manajer memilih menu Jadwal, maka manajer mengatur jadwal bagi karyawan yang telah terdaftar didalam database. Manajer dapat menambah dan menghapus jadwal karyawan didalam menu ini.
5. Ketika manajer memilih menu Laporan Kehadiran, maka manajer dapat melihat laporan dari hasil kehadiran karyawan dalam waktu harian dan bulanan.

**Tabel 13 Hasil Pengujian Kelola Karyawan**

No	Kelas	Fungsi	Skenario	Data Uji	Hasil Pengujian	Keterangan
1	Kelola Karyawan	Insert	<ul style="list-style-type: none"> <li>• Manajer memilih menu Manajer dalam Menu Utama</li> <li>• Melakukan inputan data uji</li> <li>• Menekan tombol save</li> </ul>	<p>Data Salah</p> <ul style="list-style-type: none"> <li>• Hasil capture wajah berhasil disimpan</li> <li>• ID : (kosong)</li> <li>• Nama : (kosong)</li> <li>• Jabatan : manager</li> </ul> <p>Data Benar</p> <ul style="list-style-type: none"> <li>• Hasil capture wajah berhasil disimpan</li> <li>• ID : 09011</li> <li>• Nama : Ridwan</li> <li>• Jabatan : manager</li> </ul>	<p>Data wajah disimpan tapi tidak terindetifikasi dan tidak mempunyai label</p> <p>Data wajah disimpan dan terindetifikasi sebagai label karyawan</p>	<p>Setiap label data wajah harus merujuk pada label pada data karyawan</p>

2		Delete	<ul style="list-style-type: none"> <li>• Manajer memilih menu Manajer dalam Menu Utama</li> <li>• Melakukan memilih data karyawan yang akan dihapus</li> </ul> <p>Menekan tombol del</p>	<ul style="list-style-type: none"> <li>• ID : 09011</li> <li>• Nama : Ridwan</li> <li>• Jabatan : manager</li> </ul>	<ul style="list-style-type: none"> <li>- Data karyawan dengan ID :09019 dihapus</li> <li>- semua database wajah ID : 09019 terhapus</li> </ul>	<p>Data wajah 09019 bereferensi dari ID pada karyawan tersebut sehingga jika ID hilang secara otomatis semua wajah dengan ID 09019 akan hilang.</p>
---	--	--------	--	--	--	---

## V.4 Pengujian Jadwal

### V.4.1 Skenario Pengujian Jadwal

Kelas yang diuji:

Nama kelas :

- Kelas Jadwal

Skenario Jadwal adalah:

1. Manajer melakukan registrasi dengan password.
2. Manajer memilih 3 menu yaitu Jadwal, jadwal dan laporan kehadiran.
3. Ketika manajer memilih menu Jadwal, maka muncul form yang berisi data karyawan dan fitur memilih tanggal dan manajer memilih tanggal yang akan dibuat jadwalnya
4. Setiap list karyawan akan muncul listbox dengan untuk memilih jadwal shift dalam 1 hari mulai dari pukul 07.00-15.00 dengan default “no shift”
5. Manajer melakukan perbaharuan atau *update* shift karyawan satu per satu atau mengelola terlebih dahulu semua shift karyawan dalam 1 hari kemudian *update all*.

**Tabel 14 Hasil Pengujian Jadwal**

No	Kelas	Fungsi	Skenario	Data Uji	Hasil Pengujian	Keterangan
1	Jadwal	Update	<ul style="list-style-type: none"> <li>• Melakukan inputan tanggal</li> <li>• Memilih salah satu list karyawan</li> <li>• Memilih shift pada listbox</li> <li>• Menekan tombol update pada list karyawan tersebut</li> </ul>	<ul style="list-style-type: none"> <li>- ID : 09042</li> <li>Shift : 08.00</li> </ul>	Data shift karyawan disimpan tanpa mengupdate shift karyawan lain	
		Update All	<ul style="list-style-type: none"> <li>• Melakukan inputan tanggal</li> <li>• Memilih shift untuk pada semua karyawan</li> <li>• Menekan tombol update all</li> </ul>	<ul style="list-style-type: none"> <li>- ID : 09042</li> <li>Shift : 08.00</li> <li>- ID : 09019</li> <li>Shift : no shift</li> <li>- ID : 09046</li> <li>Shift : 15.00</li> </ul>	Semua shift karyawan terupdate dan disimpan	

## **V.5 Pengujian Melihat Kehadiran**

### **V.5.1 Skenario Pengujian Melihat Kehadiran**

Kelas yang diuji:

Nama kelas :

- Kelas Melihat Kehadiran

Skenario Kelola karyawan adalah:

1. Manajer melakukan registrasi dengan password.
2. Manajer memilih 3 menu yaitu daftar karyawan, jadwal dan laporan kehadiran.
3. Ketika manajer memilih menu Melihat Kehadiran, maka muncul form Laporan
4. Manajer melihat laporan kehadiran karyawan harian dan bulanan

**Tabel 15 Hasil Melihat Kehadiran**

No	Kelas	Fungsi	Skenario	Data Uji	Hasil Pengujian	Keterangan
1	Melihat Kehadiran	Harian	<ul style="list-style-type: none"> <li>Melakukan inputan tanggal</li> </ul>	- Kehadiran tanggal 31/01/2012	Data Kehadiran seluruh karyawan tanggal 31/01/2012 tampil dengan keterangan jam masuk, keluar, dan keterangan	
		Bulanan	<ul style="list-style-type: none"> <li>Melakukan inputan bulan</li> <li>Memilih salah satu list karyawan</li> </ul>	<ul style="list-style-type: none"> <li>Kehadiran bulan Januari</li> <li>ID :09033</li> <li>Nama : Ilham</li> </ul>	Data Kehadiran ID: 09033 dalam 1 bulan januari tampil dengan keterangan dan jumlah hari kerja	

## **Bab VI Kesimpulan dan Saran**

### **VI.1 Kesimpulan**

Dari hasil implementasi dan pengujian dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi Sistem Pencatatan Kehadiran Karyawan dapat mendeteksi dan mengenali wajah karyawan terdaftar dengan tingkat akurasi 88% pada tingkat pencahayaan terang dan gelap, 90,3 % pada pencahayaan terang dan 87,3 % pada pencahayaan gelap.
2. Aplikasi Sistem Pencatatan Kehadiran Karyawan dapat menampilkan data kehadiran karyawan secara periodik meliputi keterangan jam masuk, jam keluar, keterlambatan, dan jumlah hari kerja dalam kategori laporan harian dan bulanan.
3. Aplikasi Sistem Pencatatan Kehadiran Karyawan mempermudah pengelolaan jadwal.

### **VI.2 Saran**

Saran untuk pengembangan Aplikasi Sistem Pencatatan Kehadiran Karyawan ini lebih lanjut adalah sebagai berikut:

1. Sistem pengenalan wajah dengan algoritma PCA masih berbasis 2D (dimensi). Sebaiknya dikembangkan dengan basis 3D.
2. Posisi untuk pengenalan wajah masih menggunakan 1 (satu) sudut, yaitu tegak lurus dengan wajah menghadap kamera. Sebaiknya dikembangkan sehingga posisi untuk pengenalan wajah dapat dilakukan dari berbagai sudut.
3. Pendeteksian wajah masih bisa dilakukan dengan foto. Sebaiknya dikembangkan sehingga wajah dalam foto tidak bisa dilakukan.
4. Sebaiknya dilakukan eksperimen yang lebih rinci dengan berbagai macam variasi ekspresi dan aksesoris wajah dengan kondisi lingkungan yang berbeda-beda untuk mengukur akurasi pengenalan wajah.

5. Sebaiknya aplikasi sistem Pencatatan Kehadiran Karyawan diintegrasikan dengan sistem perhitungan penggajian karyawan dan ditambahkan fitur cetak laporan.
6. Sebaiknya menggunakan database lain seperti oracle dan sql server untuk perbandingan security database dengan mysql.

## DAFTAR PUSTAKA

1. <http://computer.howstuffworks.com/facial-recognition.htm>  
(Diakses: 28/09/2011)
2. [http://en.wikipedia.org/wiki/Facial\\_recognition\\_system](http://en.wikipedia.org/wiki/Facial_recognition_system)  
(Diakses: 28 /09/2011)
3. Jurnal: Tjiharjadi, Semuil. 2011. "Pengenalan Wajah Pelanggan Toko". Universitas Kristen Maranatha: Bandung.
4. Jurnal: Rahman, Mohamad Aditya. 2010. "Sistem Pengenalan Wajah Menggunakan Webcam Untuk Absensi Dengan Metode Template Matching". Tugas Akhir: PENS-ITS Sukolilo: Surabaya.
5. Jurnal: Bayu, setia, Hendriawan, Akhwan. 2008. "Penerapan Face Recognition Dengan Metode Eigenface Dalam Intelligent Home Security". ITS: Surabaya.
6. <http://www.emgu.com> (Diakses: 28/09/2011)
7. <http://abikindo.blogspot.com/2010/06/template-matching.html>  
(Diakses: 10/10/2011)
8. [http://eprints.qut.edu.au/16289/1/Hossein\\_EbrahimpourKomleh\\_Thesis.pdf](http://eprints.qut.edu.au/16289/1/Hossein_EbrahimpourKomleh_Thesis.pdf)  
(Diakses: 11/10/2011)
9. [www.cs.otago.ac.nz/cosc453/student.../principal\\_components.pdf](http://www.cs.otago.ac.nz/cosc453/student.../principal_components.pdf)