

## Instalasi OpenCV (Open Source Computer Vision library) pada Windows 98/2000/XP

1. Dahului dengan instalasi Microsoft Visual C++ ver 6.0 (MSVC 6.0)
2. Instal DirectX 8.1 SDK dan taruh/arahkan instalasi pada directory c:\mssdk (ganti directory default c:\dxsdk menjadi c:\mssdk). Jawab Yes ketika ditanya apakah ingin menambahkan path directory pada MSVC. DirectX SDK ini dibutuhkan untuk akses informasi multimedia (video, sound, streaming dll).
3. Jalankan ipl25.exe untuk instalasi Intel Image Processing Library. Library ini dibutuhkan pada banyak aplikasi OpenCV.
4. Instal OpenCV ver 2.0 dengan menjalankan opencv\_core\_b2\_0.exe (library dasar), lalu lanjutkan dengan menjalankan opencv\_apps\_b2\_0.exe untuk instalasi aplikasi OpenCV.
5. Upgrade OpenCV dengan versi 2.1
  - Ekstrak opencv\_core\_b2.1.zip dan taruh pada directory c:\Program File\Intel\
  - Ekstrak opencv\_apps\_b2.1.zip dan taruh pada directory c:\Program File\Intel\
  - Ekstrak opencv\_patch\_b2.1.1.zip dan taruh pada directory c:\Program File\Intel\
  - Ekstrak opencv\_dll\_addon\_b2.1.1.zip dan taruh pada directory c:\Program File\Intel\
6. Perhatikan path directory pada MSVC (klik Tools>Options>Directories), dan tambahkan include dan library path:

Pada include path:

Dari MS DirectShow SDK:

```
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES
C:\MSSDK\SAMPLES\MULTIMEDIA\COMMON\INCLUDE
C:\MSSDK\INCLUDE
```

Dari OpenCV list:

```
C:\PROGRAM FILES\INTEL\OPENCV\CVAUX\INCLUDE
C:\PROGRAM FILES\INTEL\OPENCV\CV\INCLUDE
C:\PROGRAM FILES\INTEL\OPENCV\APPS\COMMON
C:\PROGRAM FILES\INTEL\OPENCV\OTHERLIBS\VLGRFMTS
```

Dari IPL:

```
C:\PROGRAM FILES\INTEL\PLSUITE\INCLUDE
```

-----

Libraries yang perlu disertakan untuk linking:

Dari MS DirectShow SDK:

```
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES\RELEASE
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES\debug
C:\MSSDK\LIB
```

Dari OpenCV:

```
C:\PROGRAM FILES\INTEL\OPENCV\LIB
```

Dari Ipl:

```
C:\PROGRAM FILES\INTEL\PLSUITE\LIB\MSVC
```

7. Copykan directory `\Intel-OpenCV2.1\aplikasi-tambahan\CvIGrFrmts` (dari CDRom) menuju directory `C:\Program Files\Intel\opencv\otherlibs\`
8. Build library tambahan pada:
  - `C:\Program Files\Intel\opencv\otherlibs\highgui`
  - `C:\Program Files\Intel\opencv\otherlibs\CvIGrFrmts`Lantas tambahkan pada path MSVC untuk include dan library dengan isian directory diatas.
9. Proses instalasi selesai, dan selanjutnya perhatikan struktur directory OpenCV pada `c:\Program File\Intel\OpenCV`

```
|
+ _DSW // Workspace file for Microsoft Visual Studio and a few Perl
utilities for statistics
|
+ CV // The library itself
| + Include // External library interface
| + _Include // Internal library interface
| + Make // Project file
| + Src // Source files
|
+ CVAux // Additional (experimental) stuff.
|
+ Docs // documentation
| + HTML // overview documentatation in HTML format
|
+ Bin // all the pre-built binaries
|
+ Lib // pre-built import and static libraries
|
+ Apps // Demo Applications
|
| + CamShiftDemo // Application - Wrapper for CamShift Tracker Filter
| + VMDemo // View Morphing demo
| + LkDemo // Lucas-Kanade Pyramid-based Point Tracker
| + HMMDemo // Hidden Markov Models Face Recognition Demo
```

```

| + StereoGR      // Stereo-based Gesture Recognition App for PointGrey
Stereo Cameras
| + Hawk          // Scripting Environment
| + Common        // CImage and CCamera classes
|
+ Filters // Direct Show filters
| + CalibFilter  // Camera Calibration filter
| + CamShift     // CamShift tracker
| + Condens      // ConDensation based tracker
| + Kalman       // Kalman filter based tracker
| + ProxyTrans   // Proxy DirectShow filter
|
+ OtherLibs
| + _Mkl         // Math Kernel Library - used for tests on matrix
functions
| + _IPL         // Image Processing Library - base library for the
OpenCV
| + HighGUI      // Simple GUI library with platform-independed
interface
| + VtGrFmts     // Library for reading/writing raster images
| + GestRec      // Experimental gesture recognition module
| + PtGrey       // Interface Module for PointGrey Stereo Camera
|
+ Tests// sources for algorithmic tests

```

Untuk aplikasi-aplikasi menggunakan video/webcam, membutuhkan library DirecShow. Build baseclasses dari direcshow, dengan open project pada MSVC yang ada pada C:\mssdk\samples\Multimedia\DirectShow\BaseClasses\baseclasses.dsw

➔ lalu build secara batch baik untuk versi release maupun debug

Bermainlah dengan contoh-contoh aplikasi bawaan OpenCV, dan buat anda familiar dengan struktur OpenCV. Baca Manual pdf yang telah disediakan, baik untuk OpenCVReference Manual maupun IPL Reference Manual.

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KRISTEN PETRA**

---

**USULAN TUGAS AKHIR**

Nama : Charles Winarto  
NRP : 26499046  
Bidang Studi : Teknologi Perangkat Lunak  
Judul Tugas Akhir : Pembuatan Aplikasi *Face Recognition* dengan menggunakan *Prinsipal Component Analysis*  
Pembimbing I : Ir. Resmana Lim, M.Eng  
Pembimbing II : Ibnu Gunawan, ST.  
Dilaksanakan : Semester Genap Tahun 2002 / 2003

Surabaya, 10 Februari 2003

Yang Mengusulkan,

Charles Winarto

Mengetahui,

Pembimbing I

Pembimbing II

Ir. Resmana Lim, M.Eng

Ibnu Gunawan, ST.

Koordinator Tugas Akhir

Cherry Galatia B., S.Si

## USULAN TUGAS AKHIR

### 1. JUDUL TUGAS AKHIR

Pembuatan Aplikasi *Face Recognition* dengan menggunakan *Prinsipal Component Analysis*

### 2. LATAR BELAKANG MASALAH

Dewasa ini semakin berkembang aplikasi – aplikasi yang ditujukan untuk mempermudah pekerjaan manusia, seperti pengenalan suara, pengenalan wajah, pelacakan wajah, pengenalan sidik jari, pengenalan plat mobil, pengenalan ekspresi wajah dan masih banyak lagi yang lain. Aplikasi – aplikasi tersebut semuanya bertujuan untuk membantu manusia dalam melaksanakan tugasnya sehari – hari dan tentunya menggunakan komputer sebagai media utama.

*Face Recognition* atau pengenalan wajah merupakan salah satu bagian penting dari proses membantu mempermudah pekerjaan manusia. Saat ini *Human Recognition* banyak sekali digunakan dan diterapkan di hampir semua segment keamanan biasanya. Selain *Face Recognition* untuk segi keamanan biasanya ada pula keamanan yang dibuat dengan menggunakan iris mata, sidik jari, kartu khusus, suara, dll. Diantara semua jenis keamanan tersebut *Face Recognition* merupakan salah satu yang paling baik karena tidak terlalu sulit untuk diterapkan, tidak memerlukan waktu yang lama untuk identifikasi, tidak memerlukan biaya tambahan hanya sebuah webcam dan tentunya sebuah komputer.

Dalam hal ini user atau manusia yang akan diidentifikasi hanya perlu berdiri menghadap ke kamera dan operator komputer akan langsung dapat mengetahui identitas individu tersebut. Hasil yang didapat kita akan dapat mengetahui apakah orang tersebut sedang dicari atau ditunggu atau yang lain.

Untuk dapat melakukan semua itu maka diperlukan sebuah teknik, ada berbagai macam teknik yang ada sekarang ini misalnya gabor filter, *Principal Component Analisis (PCA)*. Dan dengan menggunakan PCA

merupakan sebuah cara yang sangat mudah untuk dapat melakukan *Human recognition* tersebut. Dalam PCA ada sebuah database yang ditrain terlebih dahulu baru kemudian apabila ada frame yang merupakan objek yang akan diidentifikasi akan diekstrak menjadi fitur-fitur tertentu yang kemudian dibandingkan dengan “rangkuman” fitur dari database yang sudah ditrain tersebut.

Semakin banyak gambar yang kita miliki dari sebuah objek maka semakin kecil kesalahan yang akan dihasilkan. Misalnya kita memiliki database sebut saja A. Dari gambar A ini kita memiliki sisi muka dengan berbagai macam pose kemudian dari sisi kiri dan kanan juga dengan berbagai macam pose maka kemungkinan pengenalan terhadap si A ini semakin besar dari berbagai sudut kecuali dari belakang.

### **3. PERUMUSAN MASALAH DAN RUANG LINGKUP**

Dalam membuat pengenalan wajah ini perlu diperhatikan beberapa masalah antara lain :

1. Bagaimana *mentrain* atau melatih database yang kita miliki sehingga menghasilkan sebuah *average image* yang nantinya berisi fitur-fitur yang penting dari gambar – gambar yang ada didalam database yang kita miliki ?
2. Bagaimana merubah frame – frame yang kita dapatkan dari program capcam menjadi fitur-fitur tertentu yang nantinya akan kita bandingkan dengan fitur hasil *train* dari gambar-gambar database ?
3. Bagaimana cara membandingkan fitur yang didapat dari frame dengan fitur yang dihasilkan dari database ?
4. Bagaimana dari hasil yang didapat mencari gambar dari database yang sesuai dengan gambar yang sedang dicari ?

Ruang lingkup dibatasi pada :

1. Pengambilan gambar atau input wajah dari kamera (webcam) atau *AudioVideo Interleaved (AVI)*.

2. *Mentrain* gambar gambar wajah dari database menjadi fitur fitur yang penting sehingga kita dapat memiliki sebuah *average image*.
3. Merubah frame yang didapat menjadi fitur fitur tertentu sehingga siap untuk dibandingkan dengan fitur fitur dari database.
4. Membandingkan fitur dari frame dengan fitur dari database.
5. Bahasa pemrograman yang digunakan adalah Microsoft Visual C++ 6.0
6. *Library* yang digunakan adalah Microsoft Direct Show, Intel Performance Library, dan Intel Open Source Computer Vision.

#### 4. TUJUAN TUGAS AKHIR

Tujuan Tugas Akhir ini adalah untuk membuat sebuah aplikasi yang dapat melakukan pengenalan wajah seseorang sehingga nantinya seseorang yang tidak terkenal tersebut menjadi terkenal dengan database yang ada dan juga untuk pengembangan lebih lanjut mengenai *Human Recognition* yang lebih luas.

#### 5. TINJAUAN PUSTAKA

Pengenalan wajah merupakan suatu aplikasi untuk memecahkan masalah dalam pengenalan sebuah wajah yang terdapat baik dalam sebuah video ataupun kamera. Daerah wajah yang tampil baik dalam video ataupun kamera harus jelas karena data ini nantinya yang akan dipergunakan untuk pengenalan individu tersebut.

Proses secara garis besar yang terjadi dalam pengenalan wajah ini adalah :

1. Kita harus memiliki sejumlah gambar sebanyak mungkin baik dari satu individu ataupun dari jumlah individu. Gambar gambar dari database harus merupakan nama dari gambar tersebut atau nomor tertentu yang dapat dipergunakan untuk mengakses database untuk data pribadi seperti nama lengkap, alamat, no telepon, dan lain lain.
2. Kemudian dari gambar gambar tersebut, kita harus *mentrain* gambar gambar tersebut sehingga kita memiliki “rangkuman” dari gambar gambar tersebut yang nantinya dapat kita perbandingkan.



Gambar 1

Contoh Average Image yang didapat dari hasil training

3. Gambar frame yang kita dapat dari video ataupun webcam kita ubah menjadi gray kemudian kita ekstrak feature feature yang dimiliki sebuah frame tersebut sehingga kita memiliki nilai “rangkuman” juga dari frame tersebut.
4. Setelah memiliki feature yang akan dibandingkan dengan pembandingan maka kita tinggal mencari feature mana yang paling mendekati dari feature yang sedang dicari.
5. Setelah melakukan perbandingan maka kita sudah memiliki feature feature yang paling mendekati frame yang sedang kita cari dari database.



Gambar 2

Contoh gambar 5 orang yang berbeda hasil dari *eigen image* (bagian atas) dan muka hasil *train* dari database yang mendekati gambar yang dicari diatas (bagian bawah)

Proses diatas diulang sebanyak aliran frame yang masuk sehingga akan diketahui wajah wajah siapa yang ada didalam video ataupun di kamera

yang terjadi secara real time. Solusi yang lebih baik dari pengenalan wajah akan dapat dikembangkan penggunaannya dalam berbagai aplikasi seperti :

- Ijin masuk kedalam suatu ruangan dengan otomatis tanpa menggunakan operator.
- Mengidentifikasi individu yang tidak mau mengaku siapa dia.
- Video Konferensi.
- Mengidentifikasi pengguna sebuah komputer.
- Sistem keamanan yang lain.
- Dan lain lainnya.

Gambar gambar yang akan diproses semuanya dirubah menjadi *eigen image* yang kemudian akan diperbandingkan secara *Principal Component Analisis* yaitu dengan mencari kedekatan antara fitur dengan prinsip neighborhood. PCA ini akan sangat berperan sangat penting dalam pengenalan wajah ini karena proses pengenalan ini dilakukan pada proses ini dari gambar *eigen* yang dihasilkan.

## 6. METODOLOGI PENELITIAN

### 6.1 Studi Literatur

Studi literatur yang dilakukan meliputi beberapa hal antara lain :

- Mempelajari Microsoft Visual C++ 6.0.
- Mempelajari *Intel Open Source Library* dan *Image Performance Library*.
- Mempelajari cara merubah frame menjadi *eigen image*.
- Mempelajari algoritma *Principal Component Analisis*.

### 6.2 Pengumpulan data

Pengumpulan data yang dilakukan :

- Mencari gambar gambar yang cukup banyak untuk database.
- Mencari video untuk testing pengenalan sebagai input data selain dari kamera.

### 6.3 Perencanaan dan pembuatan perangkat lunak.

Perencanaan dan pembuatan sistem yang dilakukan meliputi beberapa hal antara lain :

- Perancangan *User interface* dalam Visual C++ 6.0
- Perencanaan dan pelatihan gambar dari database.
- Perencanaan dan pencarian fitur-fitur dari frame.
- Perencanaan dan pembuatan modul untuk menyimpan hasil pelatihan dari database yang sudah dilakukan agar tidak hilang.
- Perencanaan dan perbandingan antara frame *eigen image* dengan database *eigen image* dengan *Principal Component Analysis*.
- Penggabungan modul-modul tersebut menjadi sebuah sistem pengenalan wajah.

#### 6.4 Pengujian dan Analisa data.

Pengujian dan Analisa data yang dilakukan meliputi beberapa hal antara lain :

- Pengujian perangkat lunak.
- Pengujian pengolahan dan perbandingan gambar.
- Pengujian hasil pengenalan.
- Analisa output dari program.

#### 6.5 Kesimpulan

Mengevaluasi hasil pengujian dari perangkat lunak sampai didapatkan hasil yang optimal.

### 7. RELEVANSI

Relevansi dari Pengenalan wajah ini dipergunakan untuk pengembangan lebih lanjut untuk keamanan, video konferensi, validasi individu, dan lain-lain.

Relevansi yang paling besar terdapat pada keamanan dengan adanya pengenalan wajah ini diharapkan individu – individu yang tidak bertanggung jawab dapat diketahui secara dini dari kamera-kamera yang ada sehingga upaya kejahatan atau yang lainnya dapat ditekan dan dihindari.

## 8. JADWAL KEGIATAN

KEGIATAN	Bulan Ke				
	I	II	III	IV	V
Studi Literatur	v				
Pengumpulan Data	v				
Perencanaan Sistim		v			
Pembuatan Program		v	v	v	
Pengujian dan perbaikan				v	v
Penyerahan Aplikasi					v

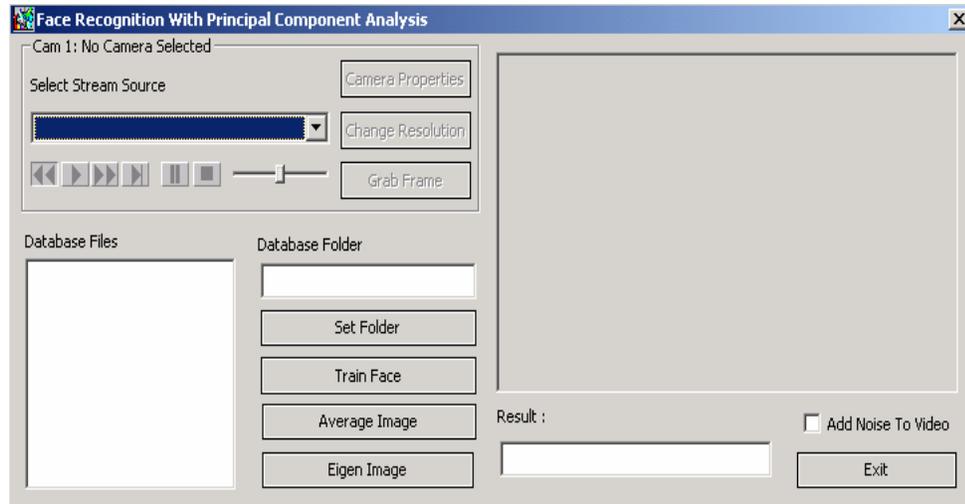
## 9. DAFTAR PUSTAKA

Forsyth, David and Ponce, Jean, "*Computer Vision – A Modern Approach*", PDF

Intel Corp., "*Open Source Computer Vision Reference Manual*", PDF.

Intel Corp. "*Image Processing Library Reference Manual*", PDF.

Pada saat pertama kali program dijalankan akan terlihat tampilan sebagai berikut :



Tampilan awal program

Pada program ini terdapat beberapa tombol, dimana tombol – tombol tersebut mempunyai fungsi sebagai berikut :

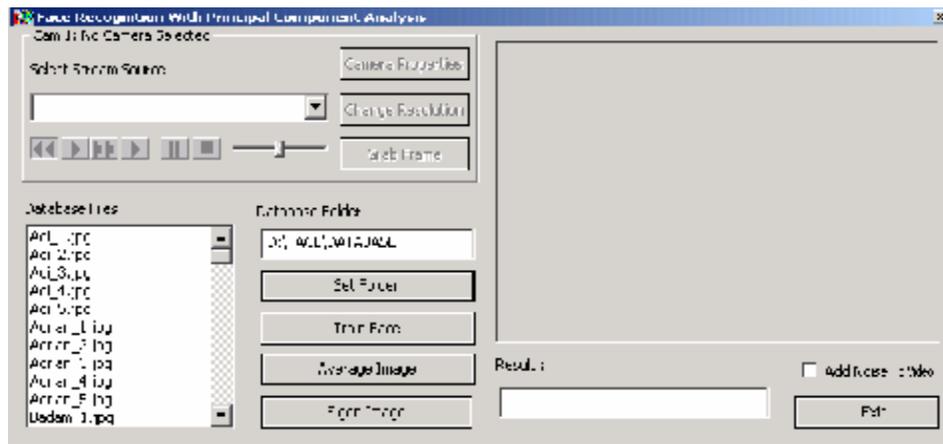
- Set Database : Berfungsi sebagai penunjuk *folder* tempat *database* wajah yang akan di-*training*-kan berada.
- Train Image : Berfungsi untuk melakukan proses *training* pada *folder* yang telah ditetapkan.
- Average Image : Berfungsi untuk menampilkan *Average Objects* yang dihasilkan setelah proses *database training*.
- Eigen Objects : Berfungsi untuk menampilkan 10 *Eigen Objects* pertama yang

dihasilkan setelah proses *database training*.

- Noise : Berfungsi untuk menambahkan noise kepada inputan video.
- Camera Properties : Berfungsi untuk memberikan informai kepada user mengenai kamera yang digunakan baik besar frame atau pun yang lainnya.
- Change Resolution : Berungsi untuk merubah resolusi dari kamera yang digunakan.
- Grab Frame : Berfungsi untuk mengambil sebuah frame dari kamera yang sedang digunakan dan menyimpannya menjadi gambar BMP.
- Backward  : Berfungsi untuk merewin video kebelakang atau ke frame yang sebelumnya.
- Forward  : Berfungsi untuk mempercepat lajunya video kedepan atau ke frame yang berada didepan daripada frame yang sedang di *play*.
- Play  : Berfungsi untuk mem*play* suatu video yang sedang di *pause* atau di-*Stop*.
- Go To The End  : Berfungsi untuk menunjuk ke frame yang paling akhir dari sebuah video.
- Pause  : Berfungsi untuk memberhentikan sementara dari video yang sedang di mainkan atau di *-play*.

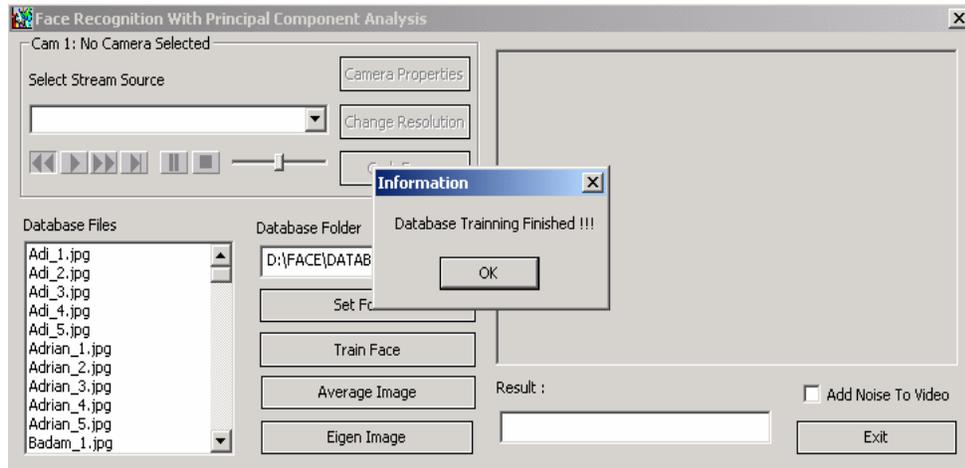
- Stop  : Berfungsi untuk memberhentikan dari video yang sedang di *-play*.

Untuk menjalankan program ini, pertama kali yang dilakukan adalah mengisi letak *folder database* wajah yang ingin di-*train*-kan pada *field database folder*. Setelah itu tekan tombol Set Database, maka pada bagian list *database files* akan muncul nama – nama file yang ada pada *folder database* tersebut.



#### Proses Pada tombol Set Folder

Setelah *folder database* disiapkan, tekan tombol Train Database untuk melakukan proses *training*. Proses *training* ini selesai jika muncul jendela informasi yang menyatakan proses *training* selesai (gambar 4.3). Setelah selesai melakukan proses *training* pada *database*, pengujian program pengenalan dapat dilakukan.



### Informasi Selesainya Proses Database Training

Pertama kali, klik Select Stream Source lalu kemudian akan muncul sebuah dialog box. Disana kemudian cari video source yang akan dipakai untuk mengukur bisa dari kamera ataupun dari file video. Setelah dipilih maka akan muncul sebuah window baru yang akan mem-*play*-kan isi dari video tersebut Hasil pengenalan wajah akan ditampilkan pada sebuah tempat disebelah kanan berupa gambar (gambar 4.5)



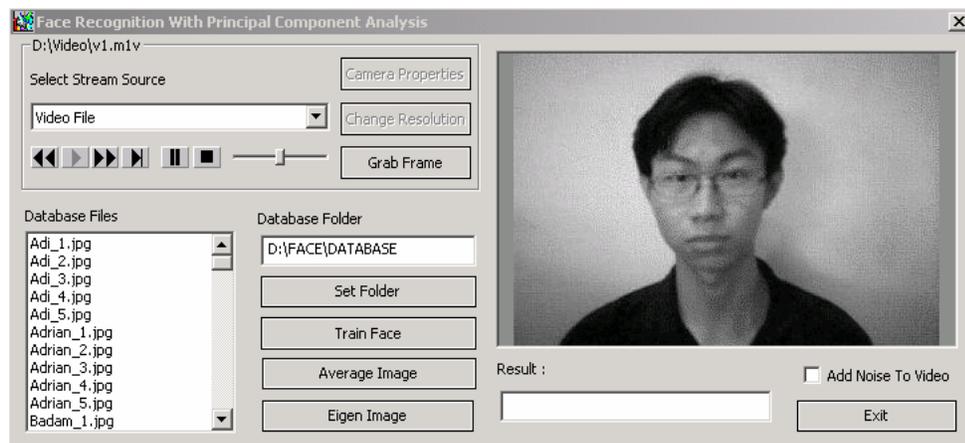
Window yang akan me-*play*-kan isi video atau kamera



Informasi pengenalan wajah di sisi sebelah kanan.



Wajah pertama dari depan



Wajah yang dikenali



Gambar kedua dari depan

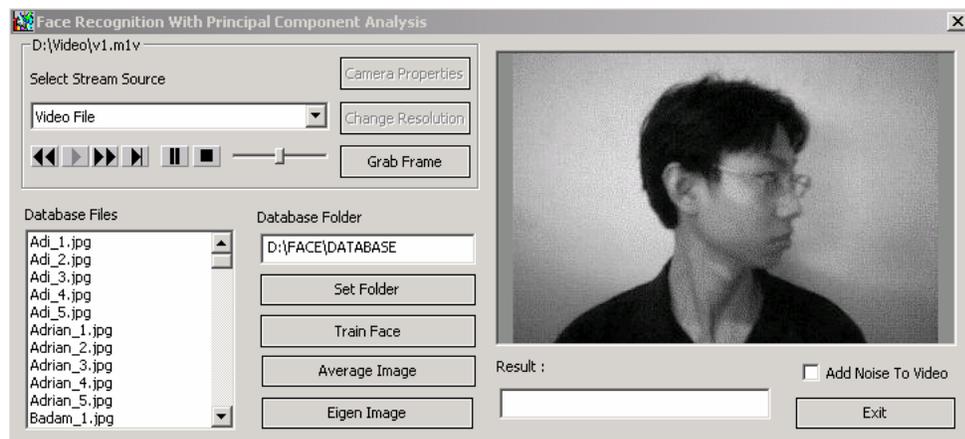


Wajah kedua yang dikenali





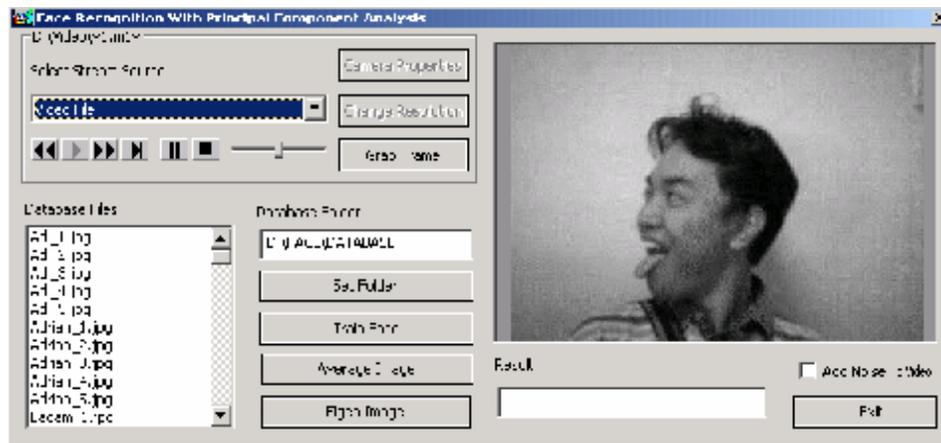
Gambar pertama dari samping



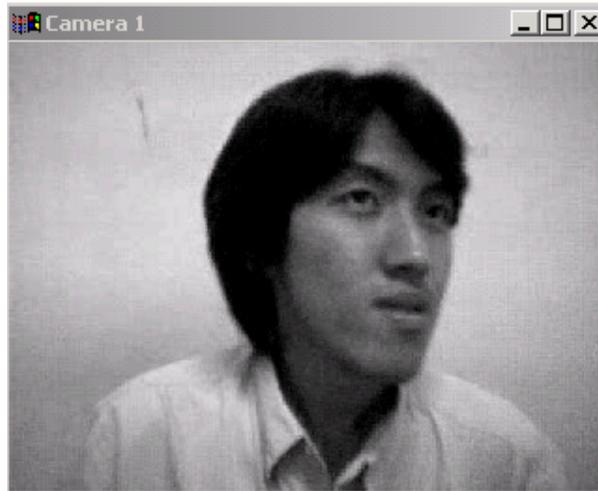
Gambar orang pertama yang dikenali



Orang kedua dari samping



Orang kedua yang dikenali dari samping



Gambar orang ketiga dari samping



Gambar orang ketiga yang dikenali dari samping