BAB IV

IMPLEMENTASI DAN PENGUJIAN

Bab ini merupakan tahap penerjemahan kebutuhan pembuatan aplikasi kedalam representasi perangkat lunak sesuai dengan hasil analisis yang telah dilakukan mencakup data-data yang digambarkan dengan tampilan. Sedangkan untuk pengujian meliputi pengujian terhadap perangkat lunak hasil implementasi.

4.1 Implementasi

Setelah melakukan analisis dan perancangan, sebuah sistem akan diimplementasikan ke dalam bentuk program menggunakan sebuah bahasa pemrograman dan sebuah kompiler. Hasil implementasi di atas kemudian akan diuji kebenarannya melalui tahapan-tahapan pengujian yang telah ditentukan. Tujuan dari implementasi ini adalah untuk mengkonfirmasikan modul program perancangan pada para pelaku sistem sehingga *user* dapat memberi masukan kepada pengembang sistem.

4.1.1 Lingkungan Implementasi

Untuk mendukung aplikasi yang akan diterapkan pada lingkungan implementasi, maka penulis menggunakan perangkat keras dan perangkat software yang menunjang dalam pengembangan aplikasi pendeteksian mata manusia dengan menggunakan algoritma *template matching* dan *eigeneye*

4.1.1.1 Perangkat Keras Yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam pengembangan aplikasi pendeteksian mata ini adalah sebagai berikut :

Tabel 4.1. Perangkat keras yang digunakan.

No	Perangkat Keras
1	*Processor* 1.66GHz
2	RAM DDR2 1GB
3	*Harddisk* 60GB
4	VGA *Onboard*
5	Monitor 14.1" WXGA

4.1.1.2 Perangkat Lunak Yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi deteksi mata adalah sebagai berikut :

Tabel 4.2. Perangkat lunak yang digunakan.

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP Professional SP2
2	Bahasa pemrograman Visual C++ 6.0	Versi 6.0
3	*Framework* .NET	Versi 2.0
4	UML Modeler	Power Designer 12 dan Rasional Rose
5	Open CV	Versi 3.1
6	DirectX SDK	November 2008

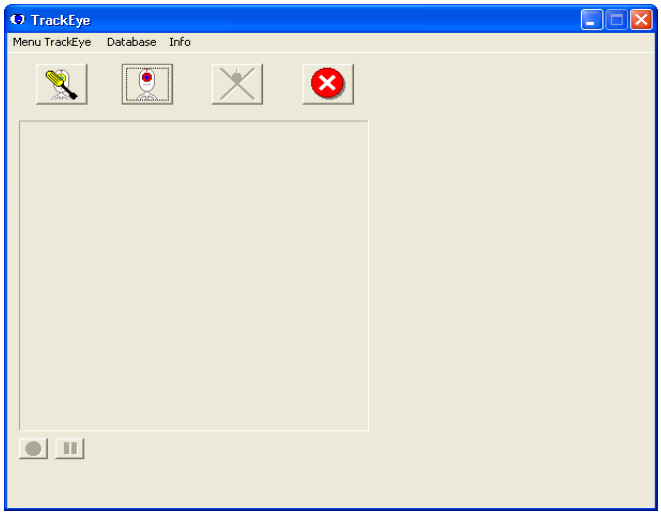
4.1.2 Batasan Implementasi

Pembatasan implementasi dimaksudkan agar ruang lingkup implementasi menjadi lebih jelas. Batasan implementasi dari aplikasi pendeteksian mata adalah sebagai berikut :

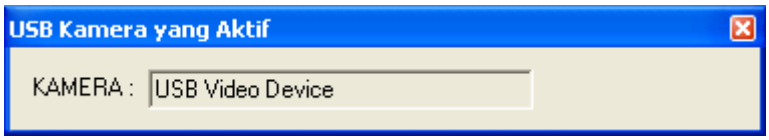
1. Perangkat lunak yang dikembangkan tidak berbasis jaringan, hanya aplikasi *desktop stand alone*.
2. Proses yang ditangani lebih terfokus pada proses pencarian mata manusia dengan menggunakan algoritma *Template Matching* dan *EigenEye*.
3. Perangkat lunak masih merupakan *prototype*.

4.1.3 Implementasi Antarmuka

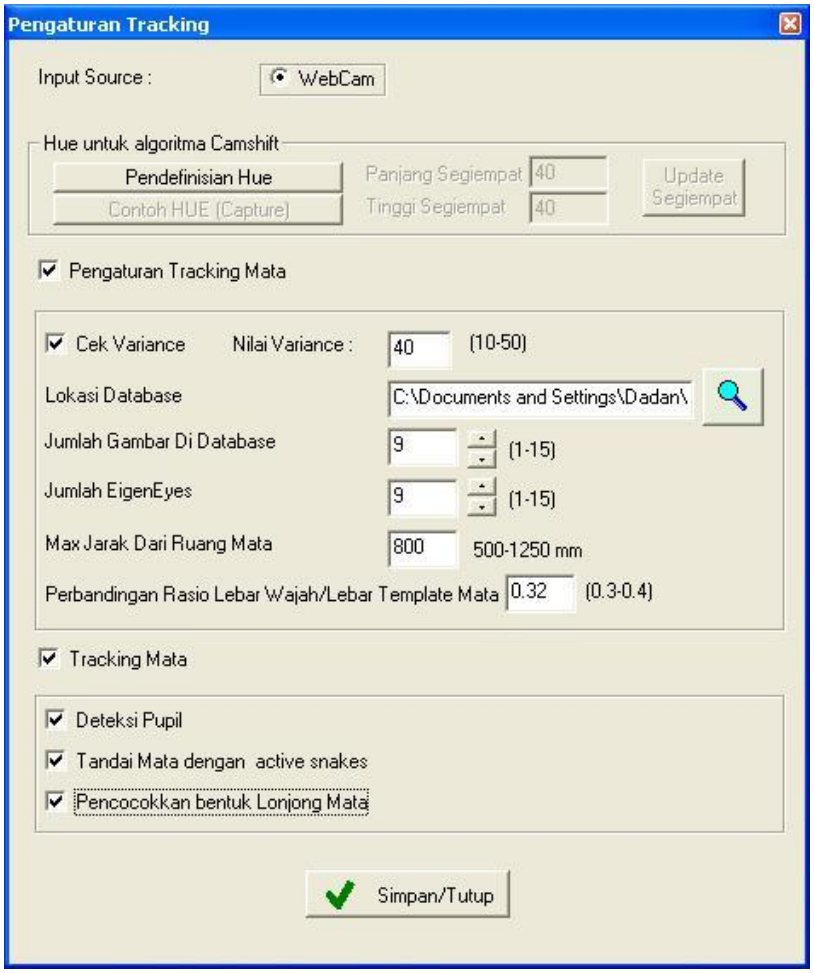
Implementasi antarmuka pengguna merupakan hasil dari perancangan antarmuka pengguna yang sebelumnya sudah dilakukan pada tahap perancangan.



Gambar 4.1. Implementasi Pengguna Antarmuka Form Main



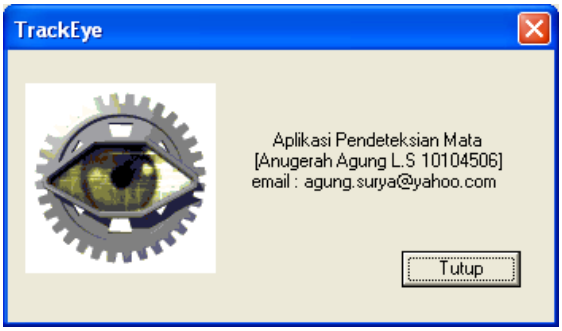
Gambar 4.2 Implementasi Pengguna Antarmuka Form Pengaturan USB kamera



Gambar 4.3 Implementasi Pengguna Antarmuka Form pengaturan tracking



Gambar 4.4 Implementasi Pengguna Antarmuka Form lihat Database



Gambar 4.5 Implementasi Pengguna Antarmuka Form Info

4.2 Pengujian Perangkat Lunak

Bab ini akan menjelaskan mengenai pengujian aplikasi Pendeteksian mata, kasus yang diujikan, dan rencana pengujian

.

4.2.1 Pengujian Aplikasi Pendeteksian Mata

Dalam pengembangan perangkat lunak, aplikasi pendeteksian mata menggunakan Microsoft Visual C++ 6.0. Yang dalam pengujian aplikasinya terdiri dari pengujian pendeteksian mata, dan pengujian terhadap respons sistem terhadap perilaku pengguna. Apakah sesuai dengan yang diharapkan atau tidak.

4.2.2 Kasus Yang Diujikan

Pengujian aplikasi pendeteksian mata hanya berfokus pada kebutuhan fungsional perangkat lunak. Kasus uji dilakukan pada proses dengan beberapa kejadian yang mungkin terjadi. Pengujiannya terdiri dari *validation testing* dan *unit testing*.

4.2.3 Pengujian Terhadap Sistem

Pengujian yang akan dilakukan adalah pada keseluruhan ssstem, untuk memeriksa apakah sistem telah bekerja dengan semestinya atau tidak, yaitu akurasi pencocokan data citra uji terhadap kelas data citra latih.

4.2.3.1 Skenario Pengujian

Dalam sistem ini, pengujian dilakukan secara subjektif. Penilaian secara subjektif ini menunjukan seberapa tepat system melakukan identifikasi citra wajah dan mata yang diberikan terhadap data citra latih tersebut tampak oleh manusia. Dengan kata lain, penelitian ini sepenuhnya tergantung pada persepsi visual manusia. Hasil penelitian kemudian dituangkan kedalam bentuk skala nilai.

Pengujian akan dilakukan dengan menguji hasil klasifikasi dan pencocokan data citra input dari user. Setelah user memasukkan data citra yang

akan diklasifikasikan akan dilihat bagaimana pembentukan kelas dan labeling dari data citra input tersebut sebagai kelas individu.

Citra masukkan yang digunakan dalam pengujian sistem ini adalah citra masukkan dari USB kamera secara *real time* yang dilakukan secara individu dengan karakteristik pose menghadap ke kamera berkisar antara nol derajat (lurus ke depan) dengan batas leher, latar belakang citra harus berwarna dan memiliki kondisi pencahayaan yang cerah.

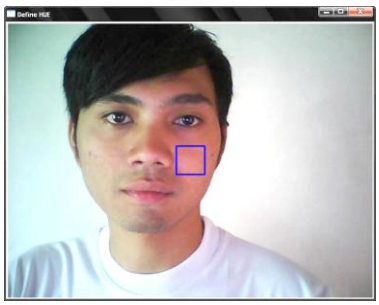
4.2.3.2 Deteksi Wajah

Tahap deteksi merupakan tahap yang signifikan untuk menentukan apakah suatu region merupakan wajah atau bukan. Adapun metode yang diterapkan pada tahap ini adalah metode Camshift. Keputusan bahwa suatu skin region merupakan wajah manusia ditentukan berdasarkan 3 persyaratan. Syarat pertama, jumlah lubang, dilakukan dengan mencari koordinat suatu skin region untuk menghasilkan gambar baru yang hanya mengandung region tersebut sisanya dibuat hitam, kemudian dihitung dengan angka Euler. Jika tidak terpenuhi, region tersebut tidak diperhitungkan lagi sehingga waktu yang diperlukan lebih singkat dibandingkan jika semua syarat terpenuhi, karena tidak perlu dilakukan proses-proses selanjutya. Akan tetapi jika hal ini terjadi pada semua skin region dalam suatu citra input berarti proses deteksi tidak berhasil.

Implementasi algoritma Camshift :

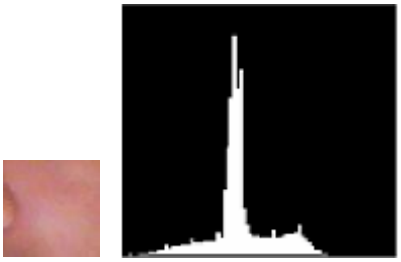
1. Suatu sample histogram HUE harus digunakan untuk dibandingkan dengan citra inputan dari kamera. Maka langkah awal untuk proses pendeteksian adalah pengguna menentukan lokasi dari hasil capture citra yang berisi bagian

dari warna kulit wajah, proses ini sangat penting untuk keakuratan dan memberikan hasil yang lebih baik pada saat proses pendeteksian wajah . Seperti gambar dibawah ini :



Gambar 4.6 Contoh pendefinsian HUE

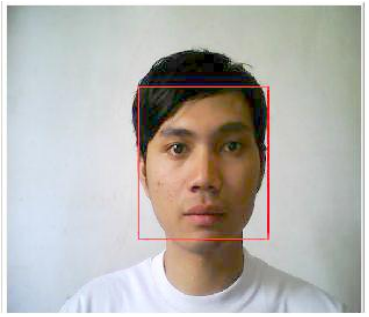
2. Setelah diperoleh contoh citra, maka akan proses selanjutnya adalah proses konversi dari warna RGB *color space* ke HSV *color space*. HUE adalah satu-satunya komponen yang menarik pada algoritma camshift.



Gambar 4.7 contoh region HUE dan histogram dari komponen HUE.

3. Suatu citra baru (proses capture warna kulit wajah) akan disimpan dan dikonversi ke HSV *color space.*
4. Inisialisasi window kalkulasi sebagai keseluruhan gambar.
5. Distribusi kemungkinan, gambar kembali diproyeksi seperti indikasi distribusi kemungkinan sebagai suatu citra gray-scale. Citra dibentuk dengan membandingkan masing-masing piksel citra source hue dengan membangun histogram HUE pada langkah ke 2.

6. model distribusi kemungkinan dan search window size menendai adanya perhitungan area wajah.



Gambar 4.8 output dari pendeteksian wajah.

7. window Kalkulasi untuk frame yang berikutnya adalah ukuran search window yang sekarang sebagai ganti keseluruhan gambaran untuk computational penyimpanan.

Tujuan dari pendeteksian wajah ini bukan hanya pendeteksian wajah secara visual. Ini berfungsi sebagai suatu sumber informasi penting untuk komponen pendeteksian mata yang di dasarkan pada Prinsipal Component Analysis, terdapat 2 informasi penting yaitu :

1. Penempatan area wajah
2. Pusat citra wajah dari keseluruhan citra, dengan begitu kita dapat mengkalkulasikan informasi lebar wajah.

Algoritma Camshift untuk pendeteksian wajah mudah diterapkan, karena dari struktur non parametic itu, penulis sudah mengamati beberapa kerugian-kerugian. Seperti efek pencahayaan ambient dan kualiatas citra yang rendah dengan menggunakan kamera yang standar.

4.2.3.3 Deteksi Mata

Implementasi algoritma pendeteksian mata pada tugas akhir ini di dasarkan pada *eigenface*. Ketika kita sedang menggunakan metode eigenface untuk pendeteksian mata di dalam area wajah, kita akan menggunakan istilah *eigeneyes*. Walaupun tujuan utama metode *eigenface* adalah untuk mencari perbandingan citra yang terbaik di dalam database, untuk suatu masukkan kita akan menggunakan metode ini untuk menggolongkan suatu masukkan citra sebagai suatu region mata atau bukan region mata, dengan cara menghitung jarak pada ruang mata.

Salah satu kelemahan metode *eigenface* adalah masalah skala. Salah satu solusi umum masalah ini adalah menggunakan database yang banyak dengan versi skala database yang asli. Solusi yang lain adalah dengan mengukur kembali masukkan citra dengan piramida gauss tetapi ketika semua skala tidak dapat digunakan selama pendeteksian, karena permasalahan kecepatan objek pada suatu jarak tertentu, tidak akan dideteksi. Jadi, skala tidak bisa secara total dipecahkan dan akan berlanjut sebagai suatu masalah pada metode *eigenface*.

4.2.3.4 Database Mata



Gambar 4.9 form lihat database

Didalam database mata citra mata harus mempunyai berbagai macam pose, baik itu mata kiri maupun mata kanan. Ciri khas dari database mata berisi citra mata untuk mata kiri dan mata kanan ketika pengguna menatap pada arah yang berbeda seperti melihat ke atas, bawah, kiri, kanan, serong bawah kiri dan kanan serta serong atas kiri dan kanan. Metode *Eigeneye* bergantung pada kondisi pencahayaan dan efek penerangan mempengaruhi sistem secara negatif. Maka untuk memperoleh hasil yang lebih baik harus dilakukan di tempat yang mempunyai kondisi pencahayaan yang baik. Salah satu keuntungan dari penggunaan citra mata sebagai training database adalah database yang kompleks seperti yang digunakan pada deteksi wajah bukanlah suatu kebutuhan atau keharusan ketika pandangan mata berubah-ubah.

Ukuran citra mata di dalam database mempunyai suatu peranan penting di dalam metode *adaptive eigeneye*. Dengan suatu citra wajah, dimana mata akan dideteksi menggunakan model wajah manusia dan menghitung factor skala dalam rumusan, database citra mata dihitung skalanya. Sehingga citra mata mempunyai

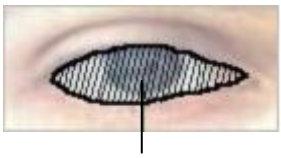
properties karakteristik area mata dan memiliki geometri. Banyaknya jumlah citra mata yang akan digunakan mempunyai peranan penting selama tahap pendeteksian berlansung.

4.2.3.5 Pengambilan Corak Mata

Sebelum tahap pengembangan, tujuan dari pembangunan aplikasi ini adalah untuk mendisain suatu sistem visi mesin yang mampu mendeteksi area mata dan kedua corak mata secara individu secara *real-time*. region sangat penting untuk penafsiran suatu citra, Suatu region di dalam sebuah citra adalah suatu kelompok piksel yang saling berhubungan dengan properties yang sama. Dibawah ini adalah contoh gambar dari suatu region mata.



Region #1



Region #2

Region #1 : pupil mata

Region #2 : Area Bola mata yang kelihatan,Region#2 berisi region#1.

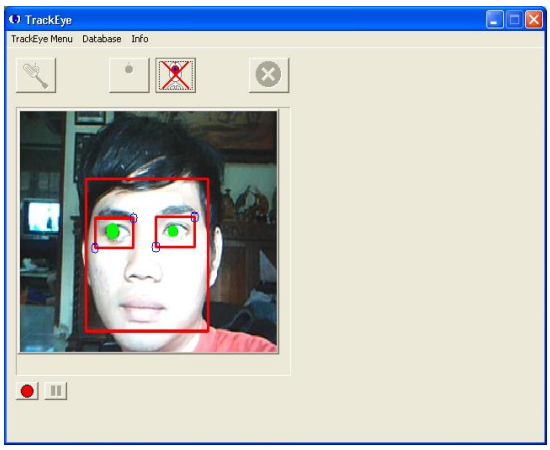
Ada dua pendekatan umum untuk menyekat suatu citra ke dalam region :

1) Segmentasi Region-based

2) *Boundary estimation* menggunakan *edge detection*

Pendekatan yang kedua yaitu hanya dengan menemukan piksel itu berada pada suatu batas region, digunakan untuk menandakan region#1 dan region#2 dalam citra.

Dibawah ini adalah contoh Output dari proses pendeteksian mata :

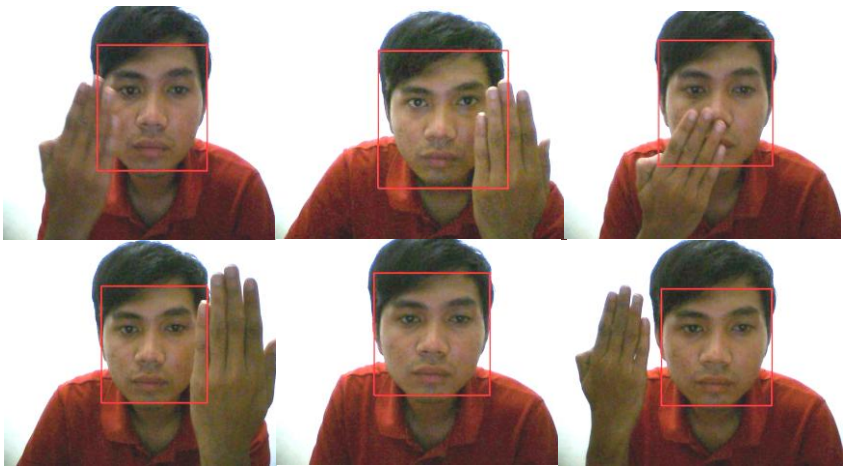


Gambar 4.10 hasil pada saat aplikasi dijalankan

Kinerja dari sistem ini adalah bergantung berdasarkan langkah masing-masing dan kinerja dari tiap langkah tersebut diakibatkan oleh hasil yang diperoleh dari langkah yang sebelumnya. Sebagai contoh, untuk track corak mata; pertama sistem harus melokalisir wajah dengan tepat, setelah lokalisasi wajah, area mata akan dicari didalam area wajah. Akhirnya setelah lokalisasi area mata, corak mata ditentukan.

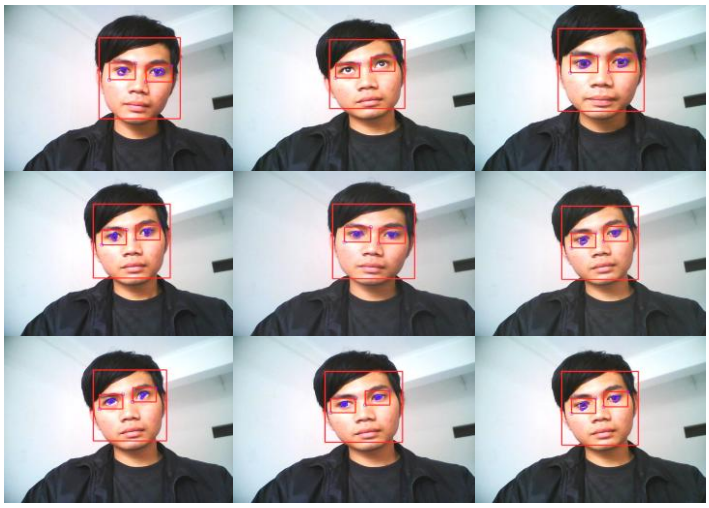
4.2.4 Pengujian Pendeteksian dengan Gangguan

CamShift search window konvergen, yaitu untuk melacak citra berwarna yang direpresentasikan dalam bentuk distribusi probabilitas citra dengan menggunakan distribusi histogram dari citra. Hanya satu area yang mempunyai propertis warna yang digambarkan dalam langkah kalibrasi yang di deteksi. Daerah yang overlap dengan wajah dan mempunyai warna kulit tidak akan diabaikan oleh CAMSHIFT.



Gambar 4.11 pengujian pendeteksian wajah dengan gangguan.

Setelah melakukan pengujian ternyata pendeteksian di dalam latar belakang yang kompleks adalah kerugian yang utama dari implementasi algoritma yang diterapkan. ketika wajah bergerak, latar belakang objek memproduksi kemungkinan distribusi warna tinggi, yangakan menyebabkan pendeteksian area wajah menjadi tidak menentu pendeteksian corak mata lebih dari satu orang atau mendeteksi corak dari mata di dalam overlap area wajah dan penggunaan latar belakang yang kompleks selama mendeteksi adalah tidak mungkin untuk suatu aplikasi *human-computer interface*. lokalisasi mata menggunakan *adaptip eigeneyes* bergantung pada informasi lebar segiempat mata yang melingkupi distribusi warna. maka jika distractors terjadi ini akan menyebabkan ukuran database salah yang akan mungkin membuat sistem gagal untuk mendeteksi area mata.



Gambar 4.12 Pendeteksian Wajah Dengan Menggunakan 3 *training image*

Akurasi pendeteksian mata dengan metode *adaptive eigeneye* adalah dependent pada banyaknya citra training karena jarak area mata pada ruang mata umumnya lebih kecil dibanding daerah wajah yaitu area mata tidak berubah secara radikal ketika diproyeksikan ke dalam ruang mata sedangkan area bukan nampak berbeda.

4.2.5 Pengujian Pendeteksian dengan Menggunakan Kacamata

Setelah dilakukan pengujian pendeteksian dengan menggunakan kaca mata, ternyata area mata terdeteksi walaupun tingkat akurasinya masih kurang yaitu 75%. Hasil pendeteksiannya dapat kita lihat pada gambar 4.25 di bawah ini.



Gambar 4.13 pendeteksian dengan menggunakan kaca mata

4.3 Pengujian Parameter Pendeteksian Mata

Pengujian parameter secara random dilakukan untuk mengetahui performansi pendeteksian yang dilakukan, untuk menentukan nilai tingkat persentase pendeteksian, hasil capture citra yang di terdeksi sebagai mata di bagi dengan jumlah citra yang di capture kemudian dikalikan 100. dengan Asumsi citra yang dicapture pada pengujian ini adalah 10 citra.

Tabel 4.3 Pengujian 1

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
15	1	2	500	0.30	60	40
30	3	4	600	0.31	75	25
40	5	6	700	0.32	70	30
35	7	8	800	0.34	80	20
30	9	10	900	0.36	60	60
50	11	12	1000	0.38	50	50
25	13	14	1100	0.39	40	40
10	15	15	1250	0.40	30	70

Tabel 4.4 Pengujian 2

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
50	2	1	500	0.30	70	30
40	4	3	600	0.31	75	25
30	6	5	700	0.32	75	25
25	8	7	800	0.34	80	20
10	10	9	900	0.36	50	50
50	12	11	1000	0.38	40	60
45	14	13	1100	0.39	30	70
30	15	15	1250	0.40	20	80

Tabel 4.5 Pengujian 3

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
40	1	15	550	0.30	55	45
35	2	14	600	0.31	65	35
20	3	13	650	0.32	75	25
10	4	12	700	0.34	85	25
30	5	11	750	0.36	50	50
30	6	10	800	0.38	40	60
40	7	9	850	0.39	30	70
30	8	8	900	0.40	25	25
10	9	7	950	0.31	80	20
35	10	6	1000	0.32	80	20
25	11	5	1050	0.34	75	25
15	12	4	1100	0.36	70	30
35	13	3	1150	0.38	50	50
45	14	2	1200	0.39	40	60
50	15	1	1250	0.40	20	80

Tabel 4.6 Pengujian 4

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
30	15	1	550	0.31	70	30
30	14	2	600	0.32	75	25
30	13	3	650	0.34	75	25
35	12	4	700	0.36	65	35
34	11	5	750	0.38	40	60
30	10	6	800	0.39	30	70
38	9	7	850	0.40	20	80
40	8	8	900	0.31	80	20
25	7	9	950	0.30	70	30
15	6	10	1000	0.31	60	40
48	5	11	1050	0.32	75	25
38	4	12	1100	0.34	50	50
30	3	13	1150	0.36	30	70
20	2	14	1200	0.38	25	75
10	1	15	1250	0.39	20	80

Tabel 4.7 Pengujian 5

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
20	1	1	950	0.31	50	50
10	2	2	1000	0.32	60	40
50	3	3	1050	0.34	65	25
50	4	4	1100	0.36	60	40
15	5	5	1150	0.38	40	60
40	6	6	1200	0.39	30	70
50	7	7	1250	0.40	20	80
15	8	8	950	0.31	75	25
30	9	9	1250	0.32	85	15
40	10	10	600	0.31	80	20
10	11	11	650	0.32	80	20
35	12	12	700	0.34	75	
10	13	13	750	0.36	55	45

50	14	14	800	0.38	40	60
25	15	15	850	0.39	30	70

Tabel 4.8 Pengujian 6

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
10	2	15	500	0.30	75	25
35	4	13	600	0.31	60	40
50	6	11	700	0.32	70	30
45	8	9	800	0.34	75	25
20	10	7	900	0.36	80	20
40	12	5	1000	0.38	50	50
35	14	3	1100	0.39	30	70
20	15	1	1250	0.40	20	80

Tabel 4.9 Pengujian 7

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
30	3	2	950	0.31	65	25
40	4	3	1000	0.32	70	30
35	8	7	1050	0.34	80	20
50	1	8	1100	0.36	60	40
40	2	1	1150	0.38	30	70
20	11	10	1200	0.39	20	80
15	12	11	1250	0.40	20	80
50	15	14	950	0.31	80	20
25	13	12	550	0.30	75	25
30	7	6	600	0.31	85	15
45	6	5	650	0.32	70	30
30	5	4	700	0.34	60	40
20	10	9	750	0.36	45	55
25	9	1	800	0.38	30	
30	14	13	850	0.39	20	

Tabel 4.10 Pengujian 8

Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
15	2	4	800	0.39	35	65
30	4	6	800	0.40	30	70
50	6	8	800	0.31	65	25
25	8	10	800	0.30	70	30
45	10	12	800	0.31	80	20
50	12	14	800	0.32	85	15
35	15	4	800	0.31	70	30
24	3	5	800	0.32	65	35
50	5	7	800	0.34	60	40
25	7	9	800	0.36	40	60
23	9	11	800	0.38	35	65
18	11	13	800	0.31	75	25
10	13	15	800	0.36	45	55
43	15	3	800	0.38	30	70
30	3	15	800	0.39	20	80

Tabel 4.11 Pengujian 9

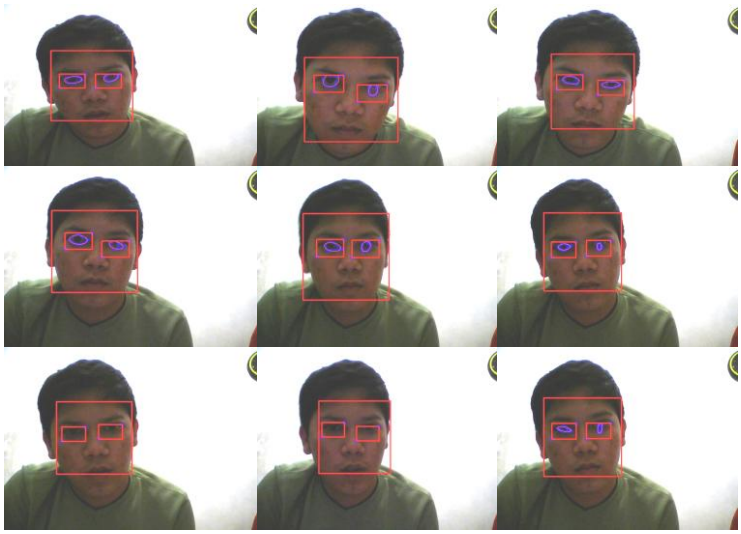
Nilai Variance	Jumlah Gambar di Database	Jumlah Eigeneyes	jarak dari Ruang Mata	Nilai rasio lebar wajah/template mata	Tingkat Pendeteksian (%)	Error Area(%)
40	2	5	650	0.35	50	50
35	6	13	550	0.37	35	65
37	8	12	1250	0.39	20	80
13	5	15	1050	0.34	55	45
18	7	4	500	0.39	20	80
50	8	8	750	0.31	75	25
50	7	9	950	0.33	80	20
30	4	9	1000	0.37	40	60
25	1	14	850	0.36	20	80
10	2	6	650	0.30	45	55
45	5	2	1250	0.40	20	80
25	7	3	1150	0.32	75	25

10	8	7	1200	0.37	30	70
50	9	10	600	0.33	80	20
35	4	1	1100	0.34	65	35

4.4 Performansi Dari Seluruh Sistem

Dari hasil pengujian parameter secara random yang sudah dilakukan di dapat parameter ideal untuk pendeteksian mata. Parameter-parameternya adalah untuk nilai variance 30, jumlah gambar di database 9 maksud dari parameter ini adalah berapa jumlah gambar yang terdapat di database yang nantinya akan di cocokkan dengan inputan citra , jumlah eigeneyes 9 yaitu jumlah pose citra secara *real time* yang akan di cocokkan dengan gambar citra training yang ada di database, sedangkan max jarak dari ruang mata idealnya 1250 (maksimal jarak yang di ijinkan dari kamera dengan objek yang akan di deteksi yaitu mata), rasio perbandingan lebar wajah/template mata 0.32 nilai parameter ini merupakan nilai rasio ideal untuk pendeteksian. Persentasi terjadinya area error atau area yang tidak terdeteksi disebabkan oleh 2 faktor yang pertama adalah faktor parameter rasio lebar wajah dan lebar *template* mata yang tidak sesuai dan pemakaian jumlah database pada citra *training* yang kecil, faktor yang kedua adalah kondisi pencahayaan meliputi background dan noise. 2 faktor tersebut sangat besar pengaruhnya dalam proses pendeteksian mata. Didalam tugas akhir ini, pendeteksian mata secara *real-time* dan *feature extracting* sistem dikembangkan. Tujuan dari tiap langkah yang dijalankan adalah suatu peranan penting untuk keluaran akhir. Salah satu keuntungan sistem adalah tidak terikat pada pemakai, Sistem memberikan hampir hasil yang sama untuk orang yang berbeda karena

keseluruhan sistem bergantung pada pendeteksian warna kulit dan pendeteksian area mata dalam region kulit. mata hampir sama dengan kebanyakan dari orang, *output* dan hasil tidak akan bertukar-tukar terlalu banyak dari orang ke orang yang berbeda hal ini karena pada dasarnya nilai hue dari setiap ras manusia sama (kecuali albino). Gambar 4.32 dibawah merupakan contoh pendeteksian dengan database mata orang yang berbeda.



Gambar 4.14 Hasil pendeteksian dengan training mata orang yang berbeda