BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem bertujuan untuk menjelaskan tentang manual modul kepada user yang akan menggunakan sistem, sehingga dapat memberikan masukan kepada pembangun sistem. Tahapan - tahapan implementasi tersebut berupa implementasi perangkat lunak, implementasi perangkat keras, implementasi Program, implementasi Antarmuka dan pengujian program.

4.1.1 Implementasi Perangkat Lunak

Implementasi Perangkat Lunak merupakan proses instalasi perangkat lunak, sehingga dapat beroperasi dengan benar. Proses implementasi perangkat lunak meliputi :

Tabel 4.1 Implementasi Perangkat Lunak

Perangkat Lunak	Keterangan
Microsoft Windows 7 Starter	Sistem Operasi
Microsoft Visual Studio 2010	Aplikasi pembangun system
Microsoft Office Visio 2007	Pemodelan system
OpenCV 2.3	Library Tambahan

4.1.2 Implementasi Perangkat Keras

Implemantasi perangkat keras merupakan realisasi dari analisis dan perancangan kebutuhan perangkat keras. Implementasi perangkat keras yang dilakukan meliputi perangkat keras yang digunakan dalam pembangunan sistem dan perangkat keras yang diperuntukan bagi komputer yang akan menggunakan aplikasi ini. Berikut ini merupakan spesifikasi perangkat keras untuk komputer dan juga webcam yang digunakan dalam pembangunan sistem :

Tabel 4.2 Implementasi Perangkat Keras Pembangun Sistem

Perangkat Keras	Spesifikasi perangkat keras
Komputer	A. Sistem operasi Windows 7 Starter Prosessor AMD Dual-Core E-350 (1.6 GHz) B. RAM 2 GB VGA AMD Radeon HD 6310 Graphics
WebCam	C. WebCam Companion 4.ink 1.3 MP 640x480

4.1.3 Implementasi Aplikasi

Implementasi aplikasi ini merupakan sebuah langkah penggunaan aplikasi sehingga dapat berjalan sesuai dengan tahap perancangan.

1. Proses objek ditangkap oleh kamera secara *realtime:*

```
capture = newCapture();
oricapture = newCapture();
  timer = newDispatcherTimer();
  timer.Tick += newEventHandler(timer_Tick);
  timer.Interval = newTimeSpan(0, 0, 0, 0, 30);
  timer.Start();

void timer_Tick(object sender, EventArgs e)

{ Image<Bgr, Byte> currentFrame = capture.QueryFrame();
```

2. Proses *grayscale:*

```
Image<Gray, Byte> grayFrame2 = currentFrame.Convert<Gray>,
```

3.	Proses deteksi mata mengantuk menggunakan metode *Haar*

	Cascade Classifier:
	privateHaarCascade DetekKi;
	DetekKi = newHaarCascade("mengantukfix.xml");
	var detectedA = grayFrame2.DetectHaarCascade(DetekKi)[0];
	foreach (var mengantuk in detectedA)
	{

4.	Memutuskan mata tersebut dalam keadaan mengantuk atau tidak :

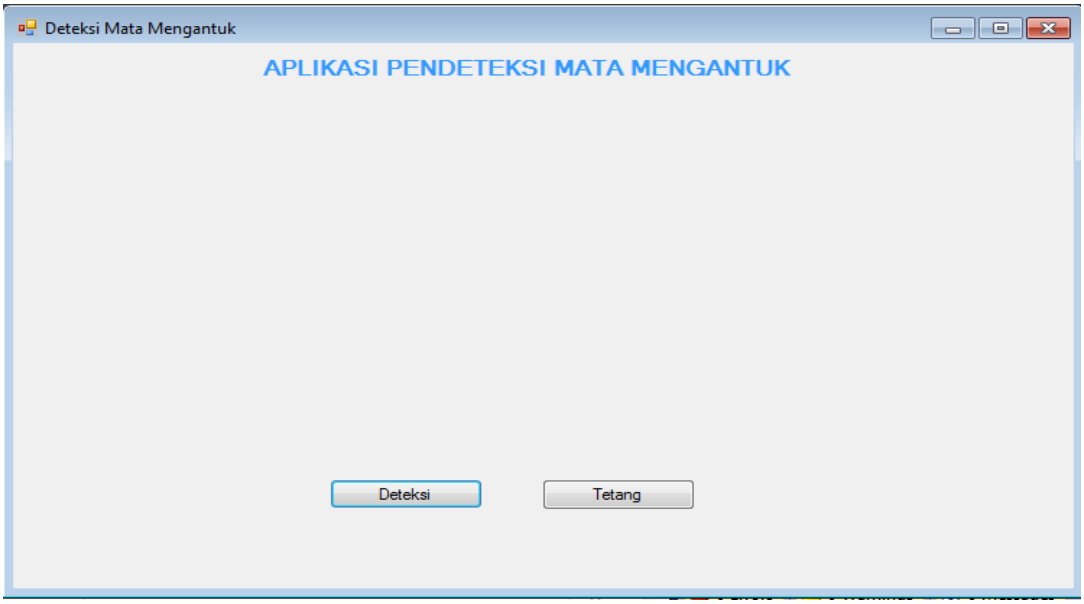
	if (mengantuknum == 0)
	{
	MCvFont font5
	=newMCvFont(Emgu.CV.CvEnum.FONT.CV_FONT_HERSHEY_
	DUPLEX,1d, 1d);
	currentFrame.Draw(String.Format("Terbuka"), ref font5,
	newPoint(5, 450), newBgr(Color.Green));
	label1.Text = "Mata Terbuka"; }
	else if (mengantuknum == 1)
	{
	MCvFont font5 =
	newMCvFont(Emgu.CV.CvEnum.FONT.CV_FONT_HERSHEY_
	DUPLEX, 1d, 1d);
	currentFrame.Draw(String.Format("mengantuk"), ref font5,

4.1.4 Implemntasi Antarmuka

Dari perancangan antarmuka yang telah dibuat pada bab sebelumnya, maka tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan. Berikut ini merupakan antarmuka yang ada pada aplikasi pendeteksi mata mengantuk.

1. Antarmuka MenuUtama

   Saat pengguna membuka aplikasi pertama kali, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka MenuUtama. Berikut ini merupakan tampilan dari antarmuka MenuUtama :



Gambar 4.1 Antarmuka MenuUtama

   Gambar di atas merupakan tampilan awal dimana sistem meminta kepada pengguna memberikan akses pada kamera agar kamera dapat berjalan sehingga dapat mengambil gambar yang dibutuhkan untuk proses selanjutnya.

2. Antarmuka untuk memasukan objek positif

   Setelah pengguna memberikan akses pada kamera maka pengguna dihadapkan pada antarmuka pendeteksi kantuk. Pada tampilan antarmuka ini terdapat 2 picturebox untuk mendeteksi kantuk. Berikut ini merupakan tampilan dari antarmuka pendekteksi Objek positif :



Gambar 4.2 Antarmuka Deteksi Objek Positif

Seperti yang terlihat pada gambar di atas, pengguna belum menghadapkan objek positif ke depan kamera.

3. Antarmuka Pendeteksi kantuk

Setelah pengguna menghdapkan objek positif pada kamera maka pengguna dihadapkan pada antarmuka pendeteksi kantuk. Pada tampilan antarmuka ini terdapat 2 picturebox untuk mendetksi kantuk. Berikut ini merupakan tampilan dari antarmuka pendekteksi kantuk :



Gambar 4.3 Antarmuka Deteksi kantuk

Seperti yang terlihat pada gambar di atas, pengguna telah menghadapkan objek positif ke depan kamera, sehingga objek bisa dideteksi mengantuk atau mata terbuka.

4.2 Pengujian

Ada dua macam pengujian yang akan dilakukan, yaitu pengujian performansi dan black box testing. Pengujian performansi yaitu dengan melakukan percobaan-percobaan dalam kondisi-kondisi tertentu seperti pengaruh posisi wajah pada saat pendeteksian, jarak wajah terhadap webcam, dan pengaruh pencahayaan terhadap deteksi mata. Pengujian pada black box testing yaitu menemukan kesalahan yang terdapat pada program.

4.2.1 Pengujian Performansi

Pengujian performansi dilakukan dengan serangkaian percobaan-percobaan dalam kondisi-kondisi tertentu yang dapat mempengaruhi keefektifan kinerja sistem pendeteksian mata mengantuk.

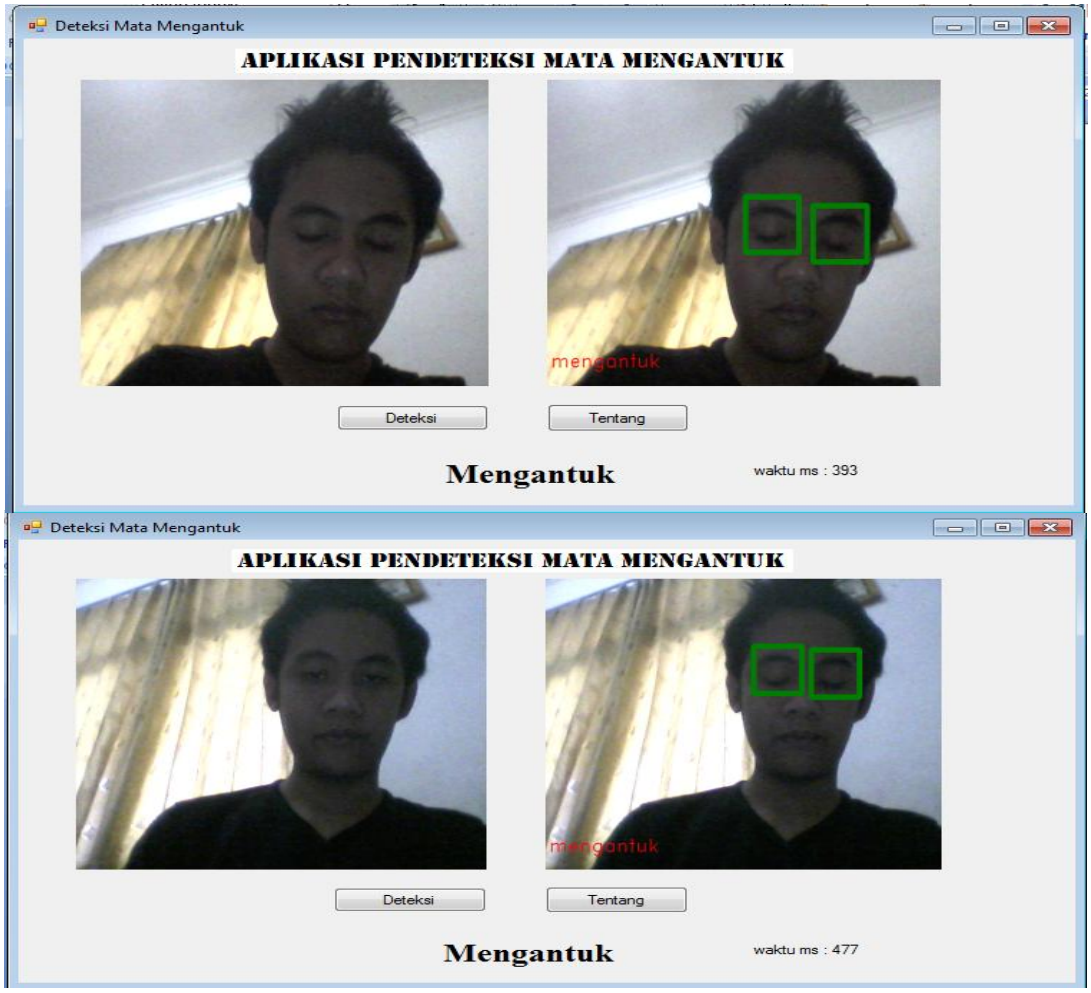
4.2.1.1 Pengujian Lama Kedipan Mata

Pada pengujian ini dilakukan untuk menguji kemampuan senstivitas *frame per second* (*fps*) webcam dalam menangkap dan mendeteksi gerakan kedipan mata yang bergerak. Uji coba terhadap deteksi pergerakan ini dilakukan dengan cara melakukan perubahan gerakan mata dari suatu posisi ke posisi yang lain dalam *window interface*.

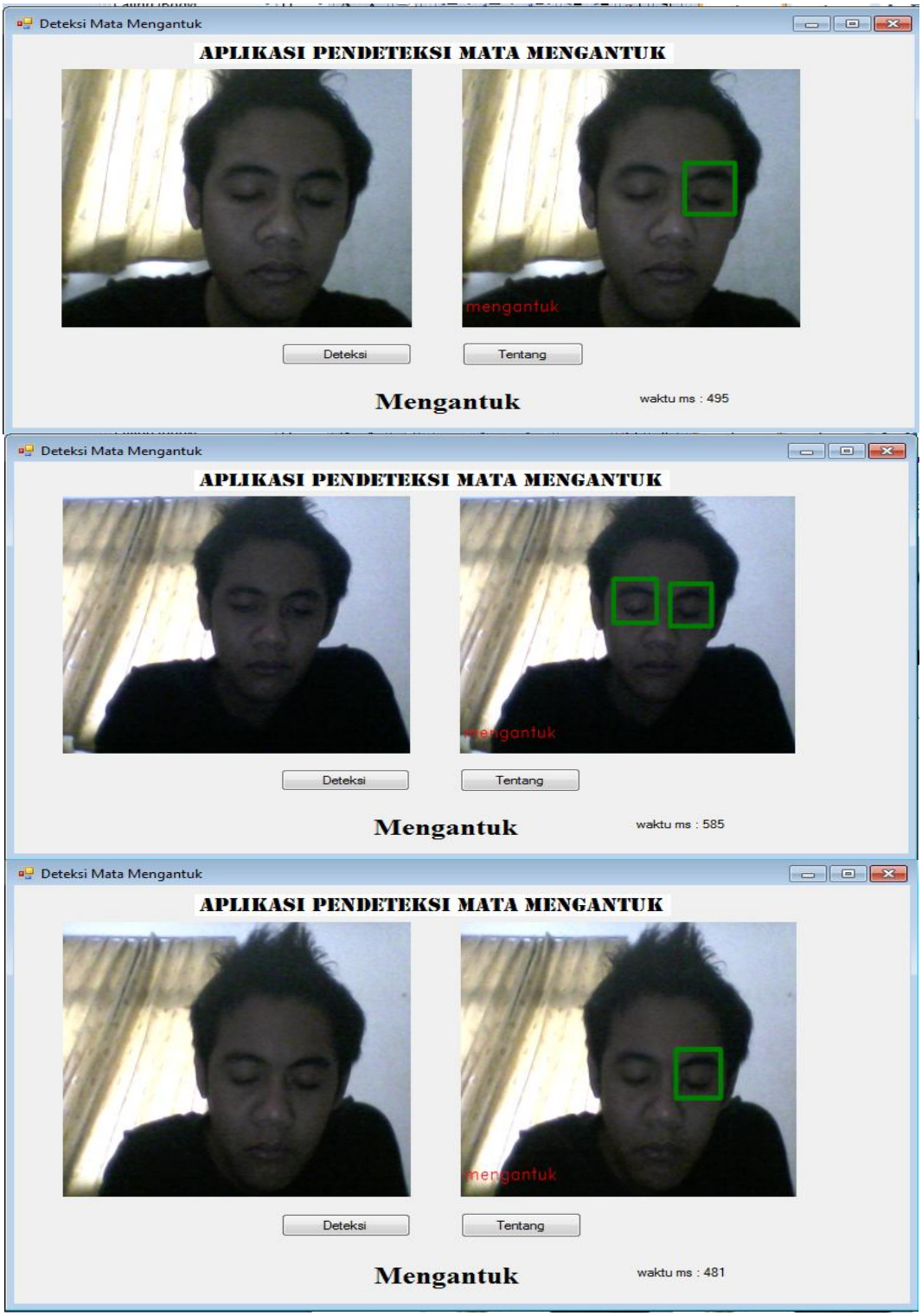
Tabel 4.3 Uji Kecepatan Kedipan Mata

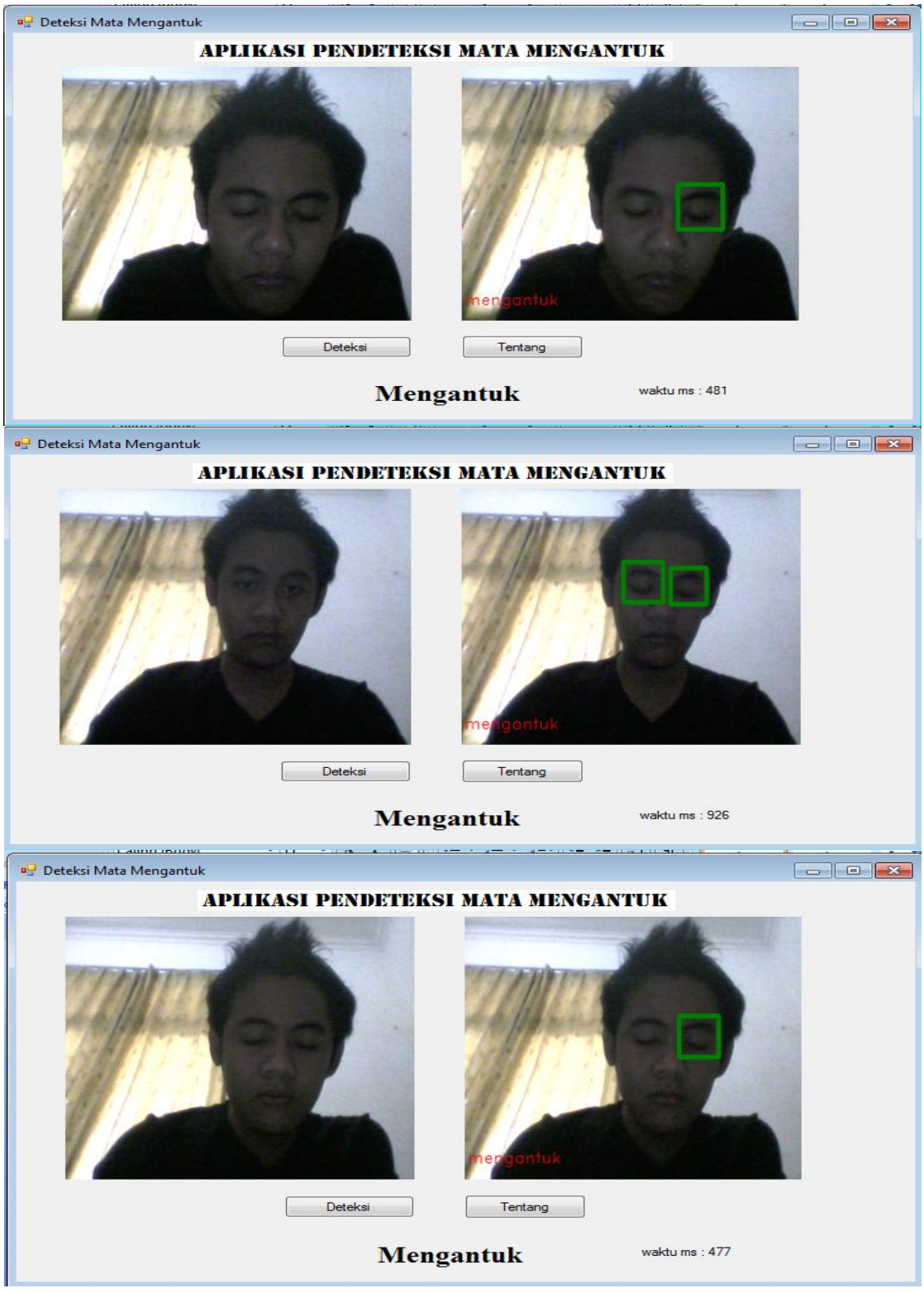
Subjek	50Ms	100Ms	200Ms	400Ms	800Ms
1	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi
2	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi
3	Tidak Terseteksi	Tidak Terseteksi	Tidak Terseteksi	Terseteksi	Terseteksi
4	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi
5	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi
6	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi

7	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
8	Tidak Terseteksi	Tidak Terseteksi	Terdeteksi	Terseteksi	Terseteksi
9	Tidak Terdeteksi	Tidak Terdeteksi	Terseteksi	Terdeteksi	Terdeteksi
10	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
Rata-rata	0%	0%	50%	100%	100%



Gambar 4.4 mengantuk


Deteksi Mata Mengantuk
APLIKASI PENDETEKSI MATA MENGANTUK
mengantuk
Deteksi
Tentang
Mengantuk
waktu ms : 495
Deteksi Mata Mengantuk
APLIKASI PENDETEKSI MATA MENGANTUK
mengantuk
Deteksi
Tentang
Mengantuk
waktu ms : 585
Deteksi Mata Mengantuk
APLIKASI PENDETEKSI MATA MENGANTUK
mengantuk
Deteksi
Tentang
Mengantuk
waktu ms : 481



Gambar 4.6 Mengantuk

Tabel dan gambar diatas menjelaskan bahwa semakin cepat gerakan yang dilakukan subyek maka sensitifitas dari sistem aplikasi mata mengantuk akan semakin rendah, begitu pula sebaliknya jika gerakan kedipan semakin lambat. Pada pengujian deteksi kedipan mata dengan parameter kecepatan kedipan mata pengguna ini ada beberapa faktor yang perlu diperhatikan antara lain: seberapa cepat kamera mengambil *frame* untuk tiap detiknya. Semakin tinggi nilai rata-rata *frame* perdetiknya maka pendeteksian kedipan mata dengan gerakan cepat pun dapat dilakukan.

Hal ini dapat diatasi dengan penggunaan kamera yang dapat mengambil *frame* sebanyak mungkin dalam satu detiknya (nilai fps tinggi). Selain itu faktor yang perlu diperhatikan berikutnya adalah seberapa baiknya processor yang di gunakan, serta seberapa besar memori komputer yang dipakai (*memory usage*) serta posisi mata harus di depan kamera.

4.2.1.2 Pengujian Deteksi Posisi Wajah

Deteksi mata mengantuk pada tugas akhir ini menggunakan proses *haar cascade*. Pengujian pada tahap ini dilakukan dengan beberapa variasi posisi sudut ke arah kiri dan ke kanan yang dibentuk oleh mata tertutup, diantaranya adalah : 0° (normal), 15°, 35°, 45°, 90° seperti ditunjukkan seperti gambar berikut :



Gambar 4.7 perubahan wajah

Pengujian ini dilakukan dengan jarak uji : 40 cm.

Tabel 4.4 Pengujian Deteksi Posisi Wajah

Subjek	Normal	15 Derajat	35 Derajat	45 Derajat	90 Derajat
1	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
2	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
3	Terseteksi	Terseteksi	Terseteksi	Tidak Terdeteksi	Tidak Terdeteksi
4	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
5	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
6	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
7	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
8	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
9	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
10	Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
Rata-rata	0%	100%	100%	100%	100%

Pada pengujian ini dapat dianalisa bahwa kondisi posisi wajah berpengaruh untuk deteksi mata mengantuk, dimana saat wajah posisi 0° (normal) proses deteksi berjalan dengan baik hingga pada posisi 35°. Sedangkan untuk posisi 45° dan 90° sistem sama sekali tidak dapat melakukan proses deteksi.

4.2.1.3 Pengujian Jarak Wajah Terhadap Webcam

Parameter yang juga sangat berpengaruh pada bagian pendeteksian wajah kali ini adalah jarak antara si pengguna dengan kamera yang digunakan. Pada pengujian kali ini hanya dilakukan pada user yang pendeteksiannya tepat pada area wajah yang tegak lurus dan ketinggiannya sejajar dengan tinggi webcam. Berikut adalah tabel pendeteksian wajah berdasarkan jarak user dengan kamera.

Tabel 4.5 Pengujian Jarak Wajah Terhadap Webcam.

Subyek	Jarak Wajah (cm)						
	20	40	60	80	100	120	>120
1	√	√	√	√	√	_	_
2	√	√	√	√	√	_	_
3	√	√	√	√	√	_	_
4	√	√	√	√	√	_	_
5	√	√	√	√	√	_	_
6	√	√	√	√	√	_	_
7	√	√	√	√	√	_	_
8	√	√	√	√	√	_	_
9	√	√	√	√	√	_	_
10	√	√	√	√	√	_	_
Rata-rata	100%	100%	100%	100%	100%	0%	0%

Keterangan :

Tanda "√" menunjukkan mata terdeteksi oleh sistem

Tanda "–" menunjukkan mata tidak terdeteksi oleh sistem

Dari pengujian diatas dapat dianalisa dan disimpulkan bahwa semakin besar jarak wajah dan semakin dekat wajah ke kamera, maka sistem deteksi mata akan bekerja lebih baik apabila dilakukan pada jarak yang agak jauh dari posisi kamera (20 cm - 100 cm). Nilai optimum dari pengujian deteksi obyek berdasarkan jarak kamera adalah 20-80 cm.

4.2.1.4 Pengaruh Cahaya Terhadap Deteksi Mata Mengantuk

Pengujian kali ini yaitu tentang pengaruh terhadap proses deteksi mata mengantuk. Dalam pengujian pengaruh pencahayaan ini menggunakan jenis lampu neon dan pijar bermerek Philips. Pada kondisi ruangan ukuran Lebar 3m, Panjang 5m dan Tinggi 3,5m dan dengan jarak 40 cm dari webcame.

Tabel 4.6 Pengujian Pengaruh Cahaya terhadap Sistem

Subyek	Jenis Lampu					
	LP 15 W	LP 40W	LP 60W	LN 5W	LN 10W	LN 15W
1	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
2	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
3	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
4	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
5	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
6	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
7	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
8	Delay 1s	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
9	Delay 1s	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
10	Delay 1s	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
Rata-rata	70%	100%	100%	100%	100%	100%

Keterangan :

LP = Lampu Pijar

LN = Lampu Neon

W = Watt

Dari pengujian di atas lampu Pijar 15 Watt mengalami delay sebanyak 3 kali, diatas 15 Watt aplikasi dapat digunakan dengan maksimal. Untuk lampu Neon ukuran 5 watt aplikasi dapat digunakan dengan baik, lebih dari itu aplikasi ini tidak mengalami masalah pada intensitas cahaya.

4.2.2 Pengujian Alpha

Pengujian *fungsional* yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Pengujian *alpha* dilakukan dengan menggunakan

metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

1. Rencana Pengujian

Rencana pengujian yang akan dilakukan dapat dilihat pada tabel berikut.

Tabel 4.7 Rencana Pengujian Aplikasi

NO	Komponen Yang Di Uji	Macam – macam Uji	Jenis Uji
1	Kamera	Memberi Akses Kepada kamera	*Black box*
		Capture video secara realtime	*Black box*
2	Interaksi Pengguna	Menekan Tombol Deteksi	*Black box*
		Menghadapkan wajah ke depan webcame	*Black box*
		Menekan tombol kembali	*Black box*
3	Deteksi	Mendeteksi mata yang mengantuk	*Black box*

2. Kasus dan Hasil Pengujian

Berikut adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode *black box* berdasarkan Tabel Rencana Pengujian yang telah dibuat.

Tabel 4.8 Kasus dan Hasil Uji

NO	Kasus/Uji	Skenario Uji	Hasil yang di harapkan	Hasil Pengujian
1	Kamera	Memberikan akses kepada kamera	Kamera bekerja setelah pengguna memilih tombol deteksi	[√]Berhasil [ ]Tidak Berhasil

			Kamera tidak bekerja ketika pengguna tidak memilih tombol deteksi	[√]Berhasil [ ]Tidak Berhasil
2	Interaksi Pengguna	Menekan Tombol Deteksi	Menampilkan Hasil webcam yang siap mendeteksi mata mengantuk	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol kembali	Menampilkan kembali ke menu utama	[√]Berhasil [ ]Tidak Berhasil
3	Deteksi Mata Mengantuk	Mendeteksi mata mengantuk	Sistem dapat mendeteksi mata mengantuk pengguna melalui webcam	[√]Berhasil [ ]Tidak Berhasil

4.2.2.1 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian alpha (fungsional) dengan kasus uji di atas dapat ditarik kesimpulan bahwa pembangunan aplikasi pendeteksi mata mengantuk tidak terdapat kesalahan proses dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.