BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

IV.1 Implementasi Sistem

Tahapan ini dilakukan setelah perancangan sistem selesai dilakukan dan selanjutnya diimplementasikan pada bahasa pemrograman yang digunakan. Tujuan implementasi sistem adalah untuk menerapkan perancangan yang telah dilakukan terhadap perangkat lunak sehingga nantinya maksud dan tujuan pembangunan perangkat lunak dapat tercapai. Berikut ini merupakan tahapan dari implementasinya :

1. Implementasi File

Tabel IV-1 Implementasi File

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <Faces_For_Training>
3    <FACE>
4      <NAME>Orang</NAME>
5  <FILE>face_Orang_2145082699.jpg</FILE>
6    </FACE>
7    <FACE>
8      <NAME>Orang</NAME>
9  <FILE>face_Orang_1067795409.jpg</FILE>
10   </FACE>
11   <FACE>
12     <NAME>Orang</NAME>
13 <FILE>face_Orang_1299679692.jpg</FILE>
14   </FACE>
15   <FACE>
16     <NAME>Orang</NAME>
17 <FILE>face_Orang_1180192084.jpg</FILE>
18   </FACE>
19 </Faces_For_Training>
```

2. Implementasi Antarmuka

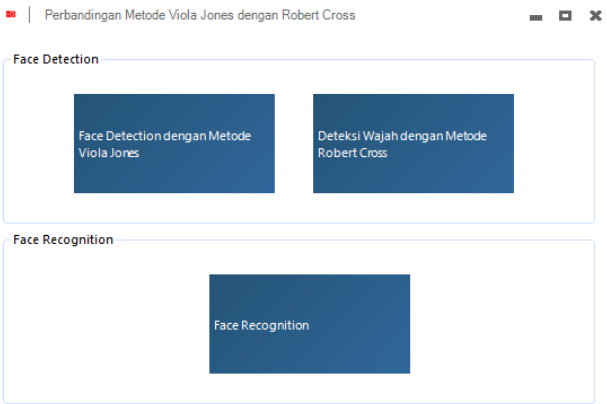
Implementasi antarmuka dibangun berdasarkan perancangan yang telah dilakukan sebelumnya. Berikut ini tabel implementasi antarmuka serta beberapa tampilan yang dapat digunakan oleh pengguna.

Tabel IV-2 Implementasi Antarmuka

No	Antarmuka	Use Case	Nama File
1	Menu Utama		MenuUtama.cs
2	Deteksi Wajah dengan Menggunakan Metode Viola Jones	Deteksi Wajah dengan Menggunakan Metode Viola Jones	ViolaJonesForm.cs
3	Deteksi Wajah dengan Menggunakan Metode Robert Cross	Deteksi Wajah dengan Menggunakan Metode Robert Cross	RobertCrossForm.cs
			FaceDetection.cs
			Camera.cs
4	Pengenalan Wajah	Pengenalan Wajah	PengenalanWajah.cs
			ClassifierTrain.cs
5	Input data wajah	Pengolahan Data Wajah	InputDataWajah.cs
			ClassifierTrain.cs

1. Antarmuka Menu Utama

   Antarmuka ini merupakan tampilan awal disaat pengguna menjalankan perangkat lunak.



Gambar IV-1 Antarmuka Menu Utama

2. Antarmuka Deteksi Wajah dengan Metode Viola Jones

   Antarmuka ini merupakan tampilan ketika pengguna memilih untuk mendeteksi wajah dengan menggunakan metode viola jones.



Gambar IV-2 Antarmuka Deteksi Wajah dengan Metode Viola Jones

3. Antarmuka Deteksi Wajah dengan Metode Robert Cross

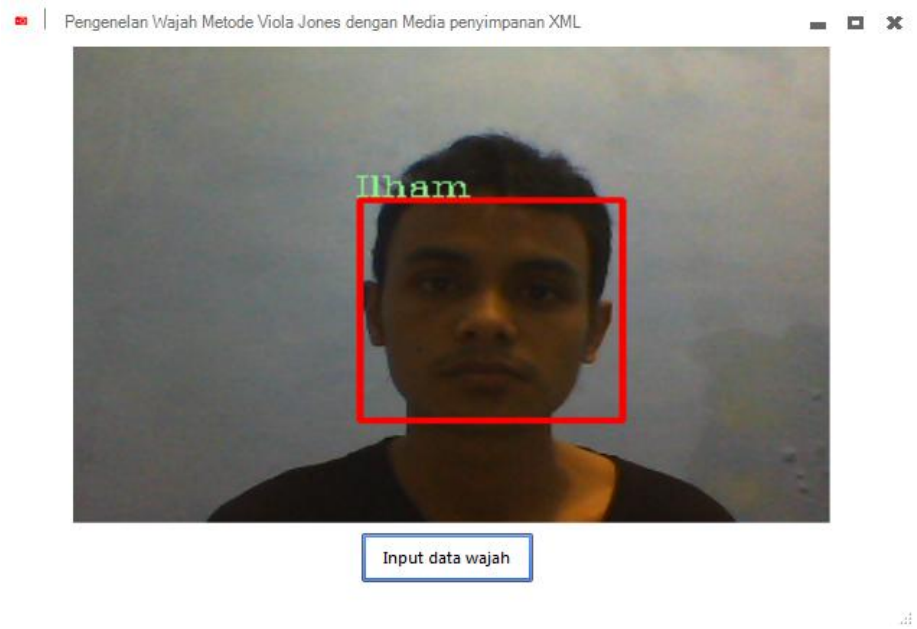
   Antarmuka ini merupakan tampilan ketika pengguna memilih untuk mendeteksi wajah dengan menggunakan metode robert cross.



Gambar IV-3 Antarmuka Deteksi Wajah dengan Metode Robert Cross

4. Antarmuka Pengenalan Wajah

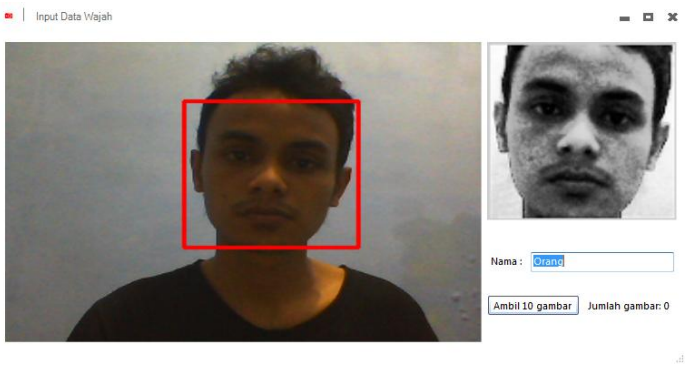
   Antarmuka ini merupakan tampilan ketika pengguna memilih menu pengenalan wajah. Ketika wajah terdeksi dan data wajah ada dalam basis data maka akan muncul nama dari wajah yang telah diinputkan kedalam basis data.



Gambar IV-4 Antarmuka Pengenalan Wajah

5. Antarmuka Input Data Wajah

   Antarmuka ini merupakan tampilan ketika pengguna menginputkan data wajah ke dalam basis data.



Gambar IV-5 Antarmuka Input Data Wajah

IV.2 Pengujian Sistem

IV.2.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada perangkat lunak ini dapat dilihat pada tabel berikut :

Tabel IV-3 Rencana Pengujian

No	Proses	Jenis Pengujian
1	Capture gambar menggunakan metode viola jones	WhiteBox
2	Capture gambar menggunakan metode robert cross	WhiteBox
3	Deteksi Wajah menggunakan metode viola jones	WhiteBox
4	Deteksi Wajah menggunakan metode robert cross	WhiteBox
5	Pengenalan Wajah	WhiteBox
6	Pengolahan Data Wajah	BlackBox
7	Hitung kecepatan deteksi wajah	BlackBox

IV.2.2 Pengujian White Box

Pengujian white box digunakan untuk mengetahui kinerja logika dari algoritma yang telah dianalisis. Ada beberapa tahapan dalam pengujian white box, berikut ini merupakan tahapan-tahapannya :

1. Mengubah *pseudocode* menjadi *flowchart*.
2. Mengubah *flowchart* menjadi *flowgraph.*
3. Tahap pengujian yang dilakukan dengan 5 cara, yaitu menghitung region, menghitung *cyclomatic complexity*, menghitung *independent path*, menggunakan *graph matriks* dan menghitung *predicate node.*

IV.2.2.1 Pengujian Algoritma Adaboost pada Metode Viola Jones

Pengujian dilakukan berdasarkan *pseudocode* pada tabel III-2.

1. Mengubah *pseudocode* menjadi *flowchart* pengujian

   Seperti yang dapat dilihat pada Gambar IV-6 *pseudocode* diubah menjadi *flowchart* pengujian.

2. Mengubah *flowchart* pengujian menjadi *flowgraph*

   Seperti yang dapat dilihat pada Gambar IV-6 *flowchart* pengujian diubah menjadi *flowgraph.*

3. Tahap Pengujian

   a. *Region*

      Terdapat 13 *region s*eperti yang dapat dilihat pada *flowgraph* Gambar IV-6.

   b. *Cyclomatic Complexity*

      V(G) = Edge – Node +2

      = 39 – 28 +2

      = 11 +2 = 13

c. *Independent Path*

Tabel IV-4 Independent Path Algoritma Adaboost

No	Path
1	Path 1 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
2	Path 2 = 1-2-4-5-6-7-8-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
3	Path 3 = 1-2-4-5-6-7-8-9-10-11-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
4	Path 4 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
5	Path 5 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
6	Path 6 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
7	Path 7 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
8	Path 8 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50
9	Path 9 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
10	Path 10 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-45-46-47-48-49-50
11	Path 11 = 1-2-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
12	Path 12 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50
13	Path 13 = 1-2-4-5-6-7-8-9-10-11-12-13-16-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36-37-40-41-42-43-44-45-46-47-48-49-50

d. *Graph matriks*

Tabel IV-5 Tabel Graph Matriks Algoritma Adaboost

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Graph Matrix
1		1	1																													1
2																																0
3				1																												0
4					1																											0
5			1			1																										1
6				1			1																									0
7					1			1																								1
8						1	1					1																				1
9								1	1																							1
10										1																						0
11									1		1																					0
12								1				1																				1
13											1		1																			0
14												1		1																		1
15													1		1																	1
16														1		1																0
17															1		1															0
18																1		1														0
19																	1		1													0
20																		1		1												0
21																			1	1	1											1
22																					1	1										0
23																						1	1									1
24																							1	1								1
25																								1	1							0
26																									1	1						0
27																										1	1					1
28																											1	1				1
29																												1	1			0
30																													1	1		1
31																														1	1	1
Jumlah																																12

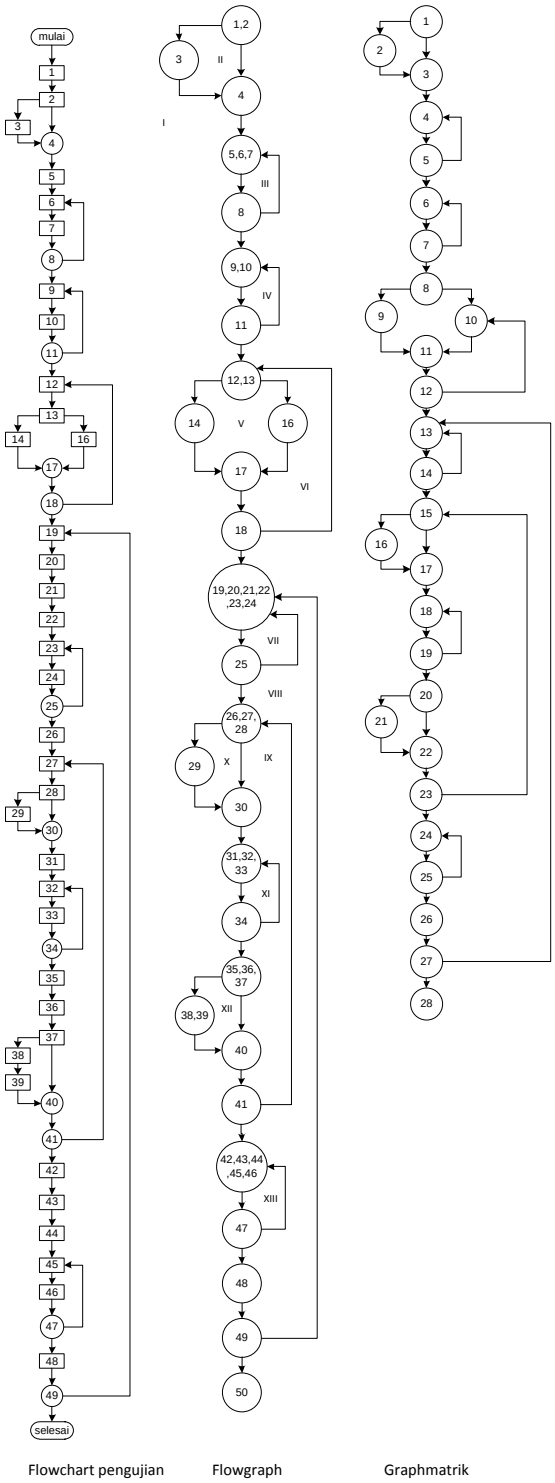
V(G) = Jumlah di Graphmatriks + 1

= 12 + 1 = 13

e. *Predicate node*

Predicate Node = 1,5,7,8,12,14,15,19,20,23,25,27

V(G) = Predicate Node + 1 = 12 + 1 = 13



Gambar IV-6 Flowchart Pengujian, Flowgraph dan Graphmatrik Algoritma Adaboost

4. Kesimpulan pengujian

   Berdasarkan hasil pengujian yang telah dilakukan maka kompleksitas dari algoritma Adaboost adalah 13.

IV.2.2.2 Pengujian Algoritma *Robert Cross Edge Detection*

1. Mengubah *pseudocode* pada tabel III-4 menjadi *flowchart* pengujian seperti yang dapat dilihat pada gambar III-7.
2. Mengubah *flowchart* pengujian menjadi *flowgraph* seperti yang dapat dilihat pada Gambar IV-7.

3. Tahap Pengujian

   a. Region

   Terdapat 4 *region* yang terdapat dalam *flowgraph* pada gambar III-7.

   b. Cyclomatic Complexity

   V(G) = Edge – Node +2

   = 9 – 7 +2

   = 2 +2

   = 4

c. Independent Path

Tabel IV-6 Algoritma *Robert Cross Edge Detection*

No	Path
1	Path 1 : 1-3-4-6-7
2	Path 2 : 1-2-3-4-6-7
3	Path 3 : 1-2-3-4-5-6-7
4	Path 4 : 1-2-3-4-6-1-2-3-4-6-7

d. GraphMatriks

Tabel IV-7 GraphMatriks Algoritma Robert Cross Edge Detection

Node	1	2	3	4	5	6	7	Graph Matrix
1		1	1					1
2			1					0
3				1				0
4					1	1		1
5						1		0
6	1						1	1
7								0
Jumlah								3

V(G) = Jumlah di Graph Matriks + 1

= 3 + 1 = 4

e. Predicate node

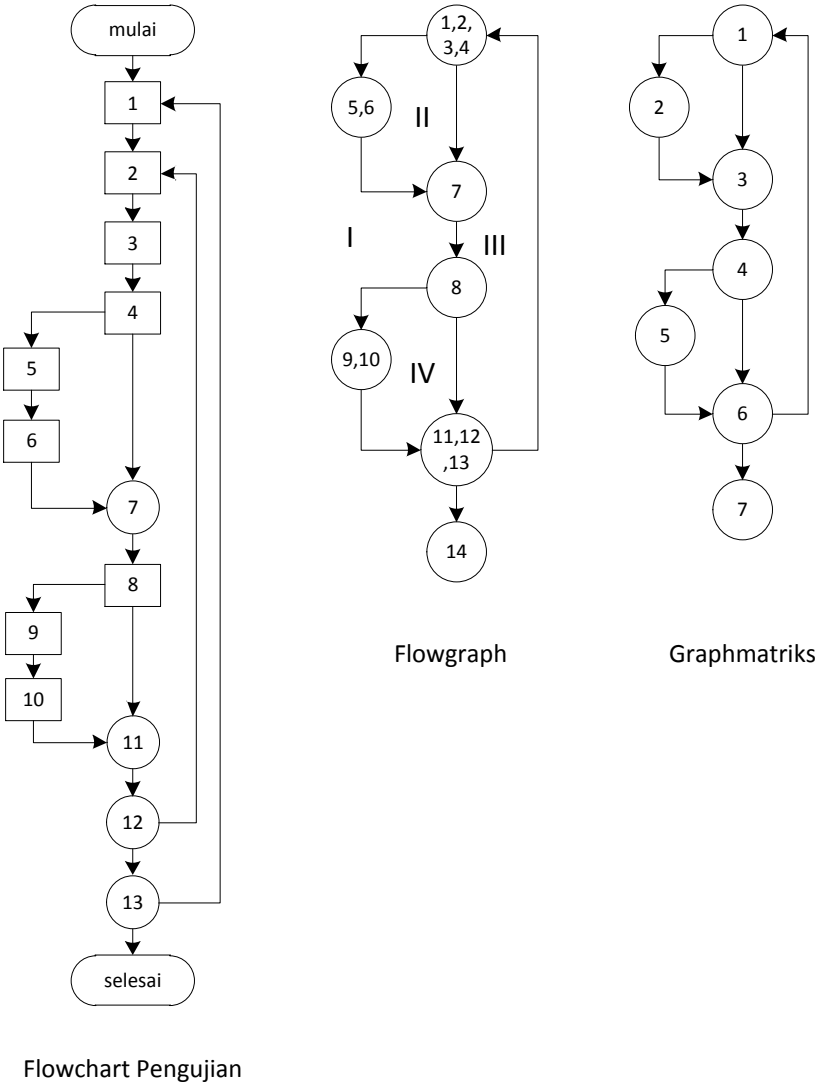
Predicate Node = 1,4,6

V(G) = Predicate Node + 1

= 3 + 1 = 4

4. Kesimpulan pengujian algoritma Robert Cross Edge Detection

   Berdasarkan hasil pengujian yang telah dilakukan di atas maka kompleksitas dari algoritma Robert Cross Edge Detection adalah 4.



Flowchart Pengujian

Flowgraph

Graphmatriks

Gambar IV-7 *Flowchart* pengujian, *Flowgraph* dan *Graphmatriks* Algoritma *Robert Cross Edge Detection*

IV.2.3 Pengujian Black Box

Pengujian *black box* bertujuan untuk mengukur kinerja dari perangkat lunak apakah fungsinya berjalan dengan baik atau tidak. Pada pengujian *black box* ini digunakan *testing performance* yang mengukur kecepatan dan ketepatan dalam proses pendeteksian wajah. Berikut pengujian *black box* yang akan dilakukan :

1. Pengujian pada tempat dengan cahaya redup

Tabel IV-8 Hasil pengujian Algoritma Viola Jones

Algoritma Adaboost Viola Jones			
No	Bentuk Wajah	Hasil	Waktu Terdeteksi
1	Muka Lonjong	terdeteksi	120
2		terdeteksi	123
3		terdeteksi	122
4		terdeteksi	123
5		terdeteksi	122
1	Muka Oval	terdeteksi	25
2		terdeteksi	29
3		terdeteksi	26
4		terdeteksi	27
5		terdeteksi	28
1	Muka Kotak	terdeteksi	120
2		terdeteksi	122
3		terdeteksi	124
4		terdeteksi	128
5		terdeteksi	127
1	Muka Bulat	terdeteksi	77
2		terdeteksi	27
3		terdeteksi	29
4		terdeteksi	28
5		terdeteksi	26

Tabel IV-9 Hasil Pengujian Algoritma Robert Cross

Algoritma Robert Cross			
No	Bentuk Wajah	Hasil	Waktu Terdeteksi
1	Muka Lonjong	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-
1	Muka Oval	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-
1	Muka Kotak	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-
1	Muka Bulat	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-

Tabel IV-10 Hasil Pengujian Pengenalan Wajah

Pengenalan Wajah			
No	Bentuk Wajah	Hasil	Waktu Terdeteksi
1	Muka Lonjong	terdeteksi	76
2		terdeteksi	66
3		terdeteksi	35
4		terdeteksi	35
5		terdeteksi	39
1	Muka Oval	terdeteksi	42
2		terdeteksi	40
3		terdeteksi	44
4		terdeteksi	38
5		terdeteksi	41
1	Muka Kotak	terdeteksi	41
2		terdeteksi	96
3		terdeteksi	70
4		terdeteksi	108
5		terdeteksi	102
1	Muka Bulat	terdeteksi	130
2		terdeteksi	137
3		terdeteksi	109
4		terdeteksi	89
5		terdeteksi	49

2. Pengujian pada tempat dengan cahaya terang

Tabel IV-11 Hasil Pengujian Algoritma Adaboost

Algoritma Adaboost Viola Jones			
No	Bentuk Wajah	Hasil	Waktu Eksekusi
1	Muka Lonjong	terdeteksi	33
2		terdeteksi	32
3		terdeteksi	40
4		terdeteksi	35
5		terdeteksi	34
1	Muka Oval	terdeteksi	25
2		terdeteksi	35
3		terdeteksi	30
4		terdeteksi	49
5		terdeteksi	38
1	Muka Kotak	terdeteksi	120
2		terdeteksi	122
3		terdeteksi	124
4		terdeteksi	128
5		terdeteksi	127
1	Muka Bulat	terdeteksi	77
2		terdeteksi	27
3		terdeteksi	29
4		terdeteksi	28
5		terdeteksi	26

Tabel IV-12 Hasil Pengujian Algoritma Robert Cross

Algoritma Robert Cross			
No	Bentuk Wajah	Hasil	Waktu Eksekusi
1	Muka Lonjong	terdeteksi	102
2		terdeteksi	90
3		terdeteksi	106
4		terdeteksi	184
5		terdeteksi	177
1	Muka Oval	terdeteksi	163
2		terdeteksi	167
3		terdeteksi	159
4		terdeteksi	186
5		terdeteksi	171
1	Muka Kotak	terdeteksi	148
2		terdeteksi	146
3		terdeteksi	158
4		terdeteksi	131
5		terdeteksi	164
1	Muka Bulat	terdeteksi	167
2		terdeteksi	106
3		terdeteksi	148
4		terdeteksi	186
5		terdeteksi	184

Tabel IV-13 Hasil Pengujian Pengenalan Wajah

Pengenalan Wajah			
No	Bentuk Wajah	Hasil	Waktu Eksekusi
1	Muka Lonjong	terdeteksi	51
2		terdeteksi	54
3		terdeteksi	53
4		terdeteksi	49
5		terdeteksi	50
1	Muka Oval	terdeteksi	42
2		terdeteksi	40
3		terdeteksi	44
4		terdeteksi	38
5		terdeteksi	41
1	Muka Kotak	terdeteksi	50
2		terdeteksi	46
3		terdeteksi	47
4		terdeteksi	50
5		terdeteksi	51
1	Muka Bulat	terdeteksi	130
2		terdeteksi	137
3		terdeteksi	109
4		terdeteksi	89
5		terdeteksi	49

IV.2.4 Kesimpulan Pengujian

Metode Viola Jones lebih unggul daripada metode Robert Cross dalam hal pendeteksian wajah. Metode Robert Cross lebih cocok untuk lingkungan dengan intensitas cahaya yang terang sedangkan metode Viola Jones cocok untuk wilayah dengan intensitas cahaya terang dan cahaya redup.