BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap pembuatan perangkat lunak, tahap lanjut dari tahap perancangan sistem. Tahap yang dilakukan untuk menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dimengerti oleh komputer serta penerapan perangkat lunak pada keadaan yang sebenarnya.

4.1.1 Implementasi Perangkat Keras

Spesifikasi perangkat keras (*hardware)* yang digunakan untuk membangun perangkat lunak pengendali pointer ini tercantum dalam tabel 4.1 berikut ini :

Tabel 4. 1 Spesifikasi perangkat keras

NO	Perangkat Keras	Spesifikasi
1	Processor	Pentium Dual Core T4200 2.00GHz
2	RAM	2.0 GB
3	Harddisk	250 GB
4	VGA card	800 MB
5	Monitor	1280 x 800
6	Kamera	Logitech HD720p

4.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membangun perangkat lunak pengendali pointer ini tercantum didalam tabel 4.2 berikut ini :

Tabel 4. 2 Spesifikasi perangkat lunak

NO	Perangkat Lunak	Keterangan
1	Windows XP Professional SP 2	Sistem Operasi

2	Microsoft Visual Studio 2010 Express	Aplikasi pembangun perangkat lunak
3	Microsoft Visio 2007	Pemodelan sistem
4	*Open Source Computer Vision* (OpenCV)	Library tambahan perangkat lunak
5	Logitech VidHD	Driver kamera/ *webcam*

4.1.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk setiap tampilan program yang dibangun dalam bentuk file code. Berikut ini adalah implementasi antarmuka untuk pengguna. Berikut ini adalah tabel implementasi antarmuka pada perangkat lunak pengendali pointer ini :

Tabel 4. 3 Tabel implementasi antarmuka

NO	Form Antarmuka	Proses	Nama File	Keterangan
1	Form Utama	Proses Pendeteksian	Aplikasi.cs	Tampilan utama dari perangkat lunak
2	Form Pengaturan	Proses Pengaturan Cahaya	Atur.cs	Tampilan pengaturan cahaya pada kamera dan pengaturan sensitifitas
3	Form Tentang Pengguan	Proses Tentang pembuat	Tentang.cs	Tampilan profil pembuat
4	Form Pesan Keluar Aplikasi	-	Pesan.cs	Tampilan pesan validasi untuk keluar dari perangkat lunak

Gambar implementasi antarmuka dapat dilihat pada bagian lampiran A.

4.2 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan – kesalahan atau kekurangan – kekurangan pada perangkat lunak yang akan diuji. Pengujian bermaksud untuk mengetahui perangkat lunak

yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.2.1 Rencana Pengujian

Rencana pengujian perangkat lunak ini dibagi menjadi 2 bagian yaitu teknik pengujian *black box* dan teknik pengujian *white box.* Pengujian *black box* berfokus pada pengujian semua perintah-perintah dan kondisi yang dieksekusi oleh perangkat lunak dan pengujian ini dilakukan oleh pembangun aplikasi. Pengujian *white box* dilakukan untuk menguji *function* pada sistem perangkat lunak dan menguji apakah *function* tersebut memenuhi kebutuhan sistem.

Strategi pengujian akan dilakukan dengan menggunakan strategi pengujian *Beta.* Strategi pengujian beta bertujuan untuk mengetahui pendapat user sebenarnya terhadap fungsi-fungsi program apakah sudah berjalan sesuai kebutuhan atau tidak. Strategi pengujian ini dilakukan dengan cara memberikan perangkat lunak kepada user yang telah ditentukan tanpa didampingi pembangun perangkat lunak.

4.2.2 Pengujian *White Box*

Pengujian *white box* digunakan untuk meyakinkan semua perintah dan kondisi pada perangkat lunak pengendali pointer agar dieksekusi secara minimal. Teknik yang digunakan yaitu *Flow Graph* yang digunakan untuk menggambarkan alur dari algoritma dan *Graph* Matriks yang digunakan untuk merepresentasikan *flow graph* dalam bentuk tabel.

4.2.2.1 Pengujian Metode Pendeteksian Gerakan Pointer

Berikut ini adalah pengujian perangkat lunak pada metode pendeteksian *image* dengan metode fisherface dari FLDA yang sebelumnya dilakukan pengenalan objek melalui OpenCVsharp dengan *function QueryFrame*. Pengujian ini dilakukan untuk pendeteksian gerakan wajah.

Berikut ini adalah *procedure* dan *function* yang akan diuji :

```
1. while (true)
2.              {
                IplImage img = cap.QueryFrame();
                CvSeq<CvAvgComp> wajah = Cv.HaarDetectObjects(img,
                fisherface, Cv.CreateMemStorage(), proses_scaling,
                MinNeighbors, HaarDetectionType.DoCannyPruning,
                MinSize);
                CvSeq<CvAvgComp> matax = Cv.HaarDetectObjects(img,
                cascade, Cv.CreateMemStorage(), proses_scaling,
                MinNeighbors, HaarDetectionType.DoCannyPruning,
                MinSize);
                CvSeq<CvAvgComp> wajahx = Cv.HaarDetectObjects(img,
                cascade2, Cv.CreateMemStorage(), proses_scaling,
                MinNeighbors, HaarDetectionType.DoCannyPruning,
                MinSize);

3. foreach (CvAvgComp mata in wajah.AsParallel())
                {
4.                img.DrawRect(mata.Rect, CvColor.Red);

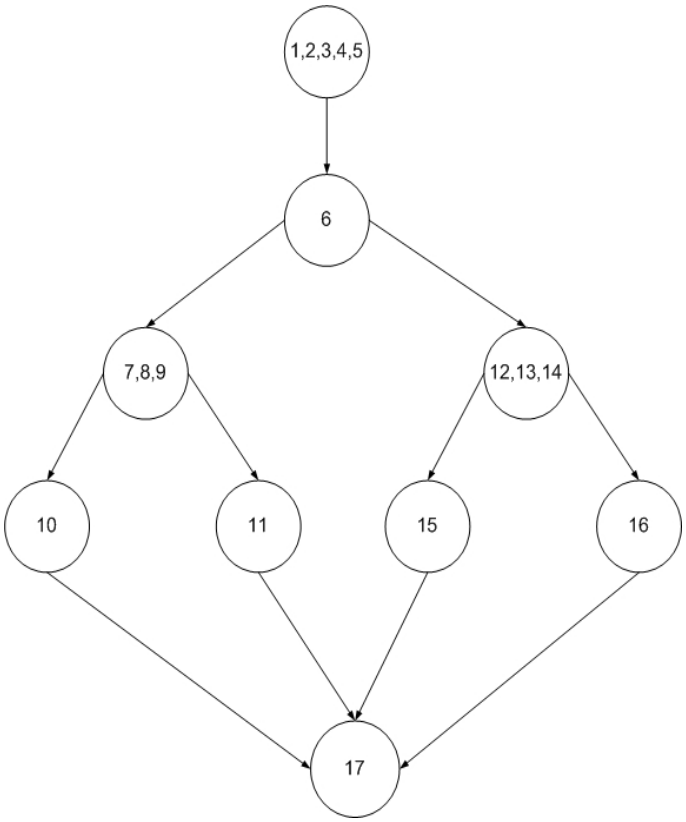
5. if (sett == true)
                {
6. lebar = this.pctCvWindow.Bounds.X + (this.pctCvWindow.Width/2);
   panjang =this.pctCvWindow.Bounds.Y+(this.pctCvWindow.Height/2);
           xt = lebar;
           yt = panjang;

   Cursor.Position = new Point(Screen.PrimaryScreen.Bounds.Width / 2,
   Screen.PrimaryScreen.Bounds.Height / 2);
   sett = false;
                }
7. x0 = 0;
8. if(Math.Abs(mata.Rect.X - xt) >=100 &&Math.Abs(mata.Rect.X - xt) <=
   190)
   {
9. if (mata.Rect.X > xt)
                      {
10.                     x0 = -3 * this.sensitif;
                      }
                      else
                      {
11.                     x0 = this.sensitif;
                      }
                   }
12.             y0 = 0;
```

```
  13. if (Math.Abs(mata.Rect.Y - yt) >= 80 && Math.Abs(mata.Rect.Y - yt)
<= 196)
  14.                {                  if (mata.Rect.Y > yt)
                                        {
  15.                                   y0 = this.sensitif;
                                        }
                                         else
                                        {
  16.                                     y0 = -3 * this.sensitif;
                                        }
                                    }
  17. Cursor.Position = new Point(Cursor.Position.X + x0,
Cursor.Position.Y + y0);
```

a. *Flow Graph* metode pendeteksian gerakan

Berikut ini adalah *flow graph* dari pendeteksian :



Gambar 4. 1 *Flow Graph* metode pendeteksian

Keterangan :

 = Node (N)

 = Anak Panah (E)

Perhitungan *white box* :

1. *Region* = 4

   Jumlah daerah tertutup atau daerah yang berada didalam node ditambah satu daerah diluar node.

2. *Cyclomatic Complexity*

   V(G) = E – N + 2

   = 11 – 9 + 2

   = 2 + 2 = 4

3. *Independent Path* = 4

   Path 1 = 1-2-3-4-5-6-7-8-9-10-17

   Path 2 = 1-2-3-4-5-6-7-8-9-11-17

   Path 3 = 1-2-3-4-5-6-12-13-14-15-17

   Path 4 = 1-2-3-4-5-6-12-13-14-16-17

Jumlah *region, Cyclomatic Complexity* dan *Independent Path* harus sama.

b. *Graph* matriks metode pendeteksian gerakan *pointer*

Satu node pada *flow graph* diwakilkan oleh satu nomor agar memudahkan perhitungan matriks. Tabel *graph* matriks dari metode pendeteksian ditunjukan pada tabel 4.4.

Tabel 4. 4 Tabel matriks *graph* pendeteksian gerakan

NODE	1	2	3	4	5	6	7	8	9	SUM(Node) – 1 (Elemen 1) 0 (Elemen Kosong
1		1								0
2			1	1						1
3					1	1				1
4							1	1		1
5									1	0
6									1	0
7									1	0
8									1	0
9										0
Jumlah										3

Keterangan :

Perhitungan nilai Cyclomatic Complexity = Nilai *Graph Matrix* + 1

= 3 + 1 = 4

4.2.2.2 Pengujian pendeteksian klik pada *pointer*

Berikut ini adalah pengujian proses klik pada *pointer mouse* yang menggunakan metode *fisherface* pada FLDA. Pengujian ini dilakukan untuk mengetahui apakah proses ini sudah memenuhi syarat *cyclomatic complexity*.

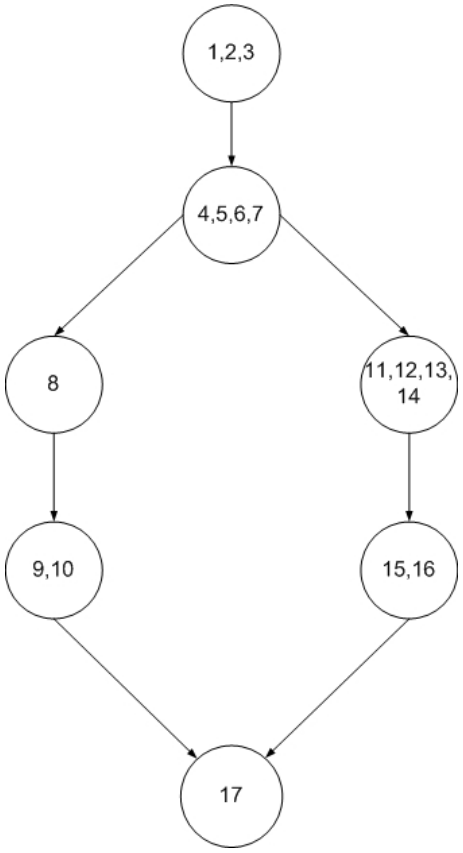
```
1. Cursor.Position = new Point(Cursor.Position.X + x0, Cursor.Position.Y +
   y0);
2. if (mata.Rect.Left > pctCvWindow.Width / 2)
   {
3. Cursor.Position = new Point(Cursor.Position.X + x0, Cursor.Position.Y +
   y0);
4. try
     {
5. IplImage mata_kanan = img.Clone();
   Cv.SetImageROI(mata_kanan, mata.Rect);
   IplImage mata_kanan2 = Cv.CreateImage(mata.Rect.Size, mata_kanan.Depth,
   mata_kanan.NChannels);
   Cv.Copy(mata_kanan, mata_kanan2, null);
```

```
        Cv.ResetImageROI(mata_kanan);
        Bitmap bitmap_mata_kanan = BitmapConverter.ToBitmap(mata_kanan2);
        proses_grayscaling = mata_kanan2.Clone();
        Bitmap GrayBm = BitmapConverter.ToBitmap(proses_grayscaling);
        img_mata_kanan.Image = bitmap_mata_kanan;
6.      FLD_kanan = Cv.Norm(proses_grayscaling, proses_thresholding);
7.          if (FLD_kanan < 2560)
                {
8.               Thread.Sleep(100);
              mouse_event(dtksi_klik_kananbwh, Cursor.Position.X,
              Cursor.Position.Y, 0, 0);
              mouse_event(dtksi_klik_kananatas, Cursor.Position.X,
              Cursor.Position.Y, 0, 0);
                 }
9.            Prev = proses_grayscaling.Clone();
                     }
10.                catch { }
                   }
                   else
                   {
11.                try
                       {
12.                        IplImage mata_kiri = img.Clone();
                                                    Cv.SetImageROI(mata_kiri,
                           mata.Rect);                          IplImage
                           mata_kiri2 = Cv.CreateImage(mata.Rect.Size,
                           mata_kiri.Depth, mata_kiri.NChannels);
                           Cv.Copy(mata_kiri, mata_kiri2, null);
                           Cv.ResetImageROI(mata_kiri);
                           Bitmap bitmap_mata_kiri =
                           BitmapConverter.ToBitmap(mata_kiri2);
                           proses_thresholding = mata_kiri2.Clone();
                           Bitmap GrayBm =
                           BitmapConverter.ToBitmap(proses_thresholding);
                           img_mata_kiri.Image = bitmap_mata_kiri;
13. FLD_kiri = Cv.Norm(proses_thresholding, Prev);
14.             if (FLD_kiri < 1200)
                           {

15.                        Thread.Sleep(100);
                           mouse_event(dtksi_klik_kiribawah, Cursor.Position.X,
                           Cursor.Position.Y, 0, 0);
                           mouse_event(dtksi_klik_kiriAtas, Cursor.Position.X,
                           Cursor.Position.Y, 0, 0);
                           }
16.                  Prev = proses_thresholding.Clone();
                        }
17.                     catch { }
                    }
                 }
```

a. Flow Graph pendeteksian klik pada *pointer*

Berikut ini adalah flow graph dari proses pendeteksian klik pada *pointer* :



Gambar 4. 2 *Flow Graph* pendeteksian klik pada *pointer*

Keterangan :

= Node (N)

= Anak Panah (E)

Perhitungan *white box* :

1. *Region* = 2

   Jumlah daerah tertutup atau daerah yang berada didalam node ditambah satu daerah diluar node.

2. *Cyclomatic Complexity*

   V(G) = E – N + 2

   = 7 – 7 + 2

   = 0 + 2 = 2

3. *Independent Path* = 2

   Path 1 = 1-2-3-4-5-6-7-8-9-10-17

   Path 2 = 1-2-3-4-5-6-7-11-12-13-14-15-16-17

Jumlah *region, Cyclomatic Complexity* dan *Independent Path* harus sama.

a. *Graph* matriks pendeteksian klik pada *pointer*

Satu node pada *flow graph* diwakilkan oleh satu nomor agar memudahkan perhitungan matriks. Tabel *graph* matriks dari metode pendeteksian ditunjukan pada tabel 4.4.

Tabel 4. 5 Tabel matrik *graph* pendeteksian klik pada *pointer*

NODE	1	2	3	4	5	6	7	SUM(Node) – 1 (Elemen 1) 0 (Elemen Kosong
1		1						0
2			1	1				1
3					1			0
4						1		0
5							1	0
6							1	0
7								0
Jumlah								1

Keterangan :

Perhitungan nilai Cyclomatic Complexity = Nilai *Graph Matrix* + 1

= 1 + 1 = 2

4.2.3 Pengujian *Black Box*

Pada pengujian black box akan dilakukan pengujian berdasarkan parameter yang ditentukan pembangun perangkat lunak untuk mengetahui apakah ada *error* pada perangkat lunak atau kemungkinan terdapat *bug* . Metode pengujian yang digunakan adalah metode *comparison testing*, Metode ini digunakan untuk membandingkan performa dari perangkat lunak. Parameter yang digunakan dalam teknik pengujian ini antara lain

1. Keadaan Cahaya
2. Perbedaan kamera
3. Kondisi *Background*
4. Bentuk wajah dan kondisi wajah
5. Nilai *Threshold* kedipan

Nilai pengujian dihitung dari jumlah terdeteksi dari setiap kali pengujian. Pengujian ini dikatakan baik apabila jumlah terdetaksinya 8-9 , diakatakan sangat baik apabila 10 kali pengujian semuanya terdeteksi, kemudian dikatakan cukup baik apabila jumlah terdeteksinya 7 kali, dan dikatakan kurang baik apabila jumlah terdeteksinya antara 1 – 6. Beikut ini adalah tabel *range* nilai bentuk wajah dan indera penglihatan.

Tabel 4. 6 Tabel range nilai bentuk objek

NO	Bentuk Objek	Range Nilai Pendeteksian	Library
1	Wajah	-9.9951568990 - 9.9896453320	Haarcascade_eya.xml
2	Indera penglihatan	-9.9892877042 - 9.9941878579	Haarcascade_frontalface_alt.xml

Pengujian 1 : Pengujian gerakan dan kedipan pada pointer mouse dilakukan pada 10 orang , kondisi pencahayaan yang baik dan menggunakan kamera Logitech 720p 3 MP dengan nilai *threshold* mata kiri 1200 dan nilai *threshold* mata kanan 2560. Nilai *threshold* pada mata diambil secara manual dengan cara menghitung rata-rata nilai perubahan frame pada posisi *region of interest* dari mata. Tabel pengujian 1 dapat dilihat pada tabel 4.6 dan 4.7.

Tabel 4. 7 Pengujian gerakan pointer

NO	Nama	Kondisi Gerakan		Keterangan
		Terdeteksi (T)	Tidak Terdeteksi(TT)	
1	Fatur (Mata kecil, Wajah bulat)	9	1	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	7	3	Pointer mengikuti gerakan wajah pengguna cukup baik
3	Willy (Matakecil, Wajah lonjong)	8	2	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	8	2	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	9	1	Pointer mengikuti gerakan wajah pengguna dengan baik
6	Kadiman (Mata besar, Wajah bulat)	10	-	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	10	-	Pointer mengikuti gerakan wajah pengguna dengan sangat baik

8	Mas Muhamad (Mata kecil, Wajah Kecil)	9	1	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	6	4	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit, Wajah bulat)	6	4	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			82% keakuratan gerakan	

Tabel 4. 8 Tabel pengujian kedipan mata

NO	Nama	Kondisi Kedipan				Keterangan
		Klik Kiri		Klik Kanan		
		Terdeteksi (T)	Tidak terdeteksi (TT)	Terdeteksi (T)	Tidak terdeteksi (TT)	
1	Fatur (Mata kecil, Wajah bulat)	7	3	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	6	4	6	4	Gerakan pointer cukup baik
3	Willy (Matakecil, Wajah lonjong)	8	2	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	8	2	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik

5	Mustopa (Mata kecil, Wajah kecil)	8	2	8	2	Pointer mengikuti gerakan wajah pengguna dengan baik
6	Kadiman (Mata besar, Wajah bulat)	9	1	9	1	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	8	2	8	2	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	9	1	8	2	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	7	3	6	4	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit, Wajah bulat)	6	4	7	3	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			Klik Kiri = 76%		Klik Kanan = 72%	

Pengujian 2 :

Pengujian gerakan dan kedipan pada pointer mouse dilakukan pada 10 orang, kondisi pencahayaan yang baik dan menggunakan kamera Standard laptop atau

kamera VGA dengan nilai *threshold* mata kiri 1200 dan nilai *threshold* mata kanan 2560. Nilai *threshold* pada mata diambil secara manual dengan cara menghitung rata-rata nilai perubahan frame pada posisi *region of interest* dari mata. Tabel pengujian 1 dapat dilihat pada tabel 4.8 dan 4.9.

Tabel 4. 9 Tabel pengujian gerakan pointer (2)

NO	Nama	Kondisi Gerakan		Keterangan
		Terdeteksi (T)	Tidak Terdeteksi(TT)	
1	Fatur (Mata kecil, Wajah bulat)	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	5	5	Pointer mengikuti gerakan wajah pengguna cukup baik
3	Willy (Matakecil, Wajah lonjong)	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik
6	Kadiman (Mata besar, Wajah bulat)	8	2	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	8	2	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik

9	Hafidz (Mata Sipit, Wajah lonjong)	5	5	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit, Wajah bulat)	5	5	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			64% keakuratan gerakan	

Tabel 4. 10 Tabel pengujian kediapan (2)

NO	Nama	Kondisi Kedipan				Keterangan
		Klik Kiri		Klik Kanan		
		Terdeteksi (T)	Tidak terdeteksi (TT)	Terdeteksi (T)	Tidak terdeteksi (TT)	
1	Fatur (Mata kecil, Wajah bulat)	6	4	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	5	5	5	5	Gerakan pointer cukup baik
3	Willy (Matakecil, Wajah lonjong)	7	3	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	6	4	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	7	3	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik

6	Kadiman (Mata besar, Wajah bulat)	8	2	8	2	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	7	3	7	3	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	8	2	7	3	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	6	4	5	5	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit, Wajah bulat)	5	5	6	4	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			Klik Kiri = 65%		Klik Kanan = 62%	

Pengujian 3 :

Pengujian gerakan dan kedipan pada pointer mouse dilakukan pada 10 orang, kondisi pencahayaan yang baik dan menggunakan kamera Logitech 720p 3MP dengan nilai *threshold* mata kiri 2200 dan nilai *threshold* mata kanan 3560. Nilai *threshold* pada mata diambil secara manual dengan cara menghitung rata-rata nilai

perubahan frame pada posisi *region of interest* dari mata. Tabel pengujian 1 dapat dilihat pada tabel 4.10 dan 4.11.

Tabel 4. 11 Tabel pengujian gerakan pointer (3)

NO	Nama	Kondisi Gerakan		Keterangan
		Terdeteksi (T)	Tidak Terdeteksi(TT)	
1	Fatur (Mata kecil, Wajah bulat)	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	4	6	Pointer mengikuti gerakan wajah pengguna cukup baik
3	Willy (Matakecil, Wajah lonjong)	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
6	Kadiman (Mata besar, Wajah bulat)	7	3	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	7	3	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	4	6	Pointer mengikuti gerakan wajah pengguna kurang baik

10	Shinta (Mata Sipit, Wajah bulat)	4	6	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			54% keakuratan gerakan	

Tabel 4. 12 Tabel pengujian kedipan (3)

NO	Nama	Kondisi Kedipan				Keterangan
		Klik Kiri		Klik Kanan		
		Terdeteksi (T)	Tidak terdeteksi (TT)	Terdeteksi (T)	Tidak terdeteksi (TT)	
1	Fatur (Mata kecil, Wajah bulat)	5	5	4	6	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	4	6	4	6	Gerakan pointer cukup baik
3	Willy (Matakecil, Wajah lonjong)	7	3	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	5	5	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	7	3	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik

6	Kadiman (Mata besar, Wajah bulat)	7	3	7	3	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	6	4	6	4	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	7	3	6	4	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	6	4	4	6	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit, Wajah bulat)	5	5	5	5	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			Klik Kiri = 59%		Klik Kanan = 52%	

Pengujian 4 :

Pengujian gerakan dan kedipan pada pointer mouse dilakukan pada 10 orang, kondisi pencahayaan yang baik dan menggunakan kamera standar laptop atau kamera VGA dengan nilai *threshold* mata kiri 2200 dan nilai *threshold* mata kanan 3560. Nilai *threshold* pada mata diambil secara manual dengan cara

menghitung rata-rata nilai perubahan frame pada posisi *region of interest* dari mata. Tabel pengujian 1 dapat dilihat pada tabel 4.12 dan 4.13.

Tabel 4. 13 Tabel pengujian gerakan pointer (4)

NO	Nama	Kondisi Gerakan		Keterangan
		Terdeteksi (T)	Tidak Terdeteksi(TT)	
1	Fatur (Mata kecil, Wajah bulat)	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	3	7	Pointer mengikuti gerakan wajah pengguna cukup baik
3	Willy (Matakecil, Wajah lonjong)	4	6	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	4	6	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
6	Kadiman (Mata besar, Wajah bulat)	6	4	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
7	Yana (Mata besar, Wajah bulat)	6	4	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	4	6	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit,	3	7	Pointer mengikuti gerakan wajah

	Wajah bulat)			pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			45% keakuratan gerakan	

Tabel 4. 14 Tabel pengujian kedipan (4)

NO	Nama	Kondisi Kedipan				Keterangan
		Klik Kiri		Klik Kanan		
		Terdeteksi (T)	Tidak terdeteksi (TT)	Terdeteksi (T)	Tidak terdeteksi (TT)	
1	Fatur (Mata kecil, Wajah bulat)	4	6	3	7	Pointer mengikuti gerakan wajah pengguna dengan baik
2	Gumara (Berkacamata, Wajah bulat)	3	7	3	7	Gerakan pointer cukup baik
3	Willy (Matakecil, Wajah lonjong)	6	4	4	6	Pointer mengikuti gerakan wajah pengguna dengan baik
4	Harry (Mata kecil, Wajah lonjong)	4	6	4	6	Pointer mengikuti gerakan wajah pengguna dengan baik
5	Mustopa (Mata kecil, Wajah kecil)	6	4	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
6	Kadiman (Mata besar, Wajah bulat)	6	4	6	4	Pointer mengikuti gerakan wajah pengguna dengan sangat baik

7	Yana (Mata besar, Wajah bulat)	5	5	5	5	Pointer mengikuti gerakan wajah pengguna dengan sangat baik
8	Mas Muhamad (Mata kecil, Wajah Kecil)	6	4	5	5	Pointer mengikuti gerakan wajah pengguna dengan baik
9	Hafidz (Mata Sipit, Wajah lonjong)	4	6	3	7	Pointer mengikuti gerakan wajah pengguna kurang baik
10	Shinta (Mata Sipit, Wajah bulat)	4	6	4	6	Pointer mengikuti gerakan wajah pengguna kurang baik
Tingkat Keakuratan (Jumlah T /100 * 100)			Klik Kiri = 48%		Klik Kanan = 42%	