BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi sistem merupakan tahap menterjemahkan perancangan berdasarkan hasil analisis sistem. Pada tahap implementasi sistem ini, diharapkan sistem yang telah dirancang siap untuk dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan tujuan yang diharapkan.

Dalam implementasi sistem terdiri dari implementasi perangkat keras, implementasi perangkat lunak, implementasi antarmuka dan implementasi basis data. Dalam implementasi ini diharapkan dapat memenuhi persyaratan agar sistem yang dibangun dapat dijalankan dengan baik.

4.1.1 Implementasi Perangkat Keras

Perangkat keras pembangun menjelaskan perangkat lunak yang digunakan untuk membangun aplikasi pendeteksi ras kucing. Adapun perangkat lunak yang digunakan untuk membangun aplikasi pendeteksi ras kucing adalah sebagai berikut:

1 Processor Intel(R) Atom(TM) 1,7 GHz.
2 RAM 2 GB.
3 Harddisk 250 MB.
4 VGA ATI Radeon HD 4600 Series.

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak pembangun menjelaskan perangkat lunak yang digunakan untuk membangun aplikasi pendeteksi ras kucing. Adapun perangkat lunak yang

digunakan untuk membangun aplikasi pendeteksi ras kucing adalah sebagai berikut:

1 Sistem Operasi Windows 7
2 Eclips Helios
3 OpenCV library
4 Adobe Photoshop CS3

4.1.3 Implementasi Antarmuka

Pada bagian implementasi antarmuka berisi pemaparan setiap tampilan aplikasi pendeteksi ras kucing yang dibangun beserta *file* xml sebagai *layout* dari masing-masing antarmuka. Adapun implementasi antarmuka yang digunakan untuk membangun aplikasi pendeteksi ras kucing adalah sebagai berikut:

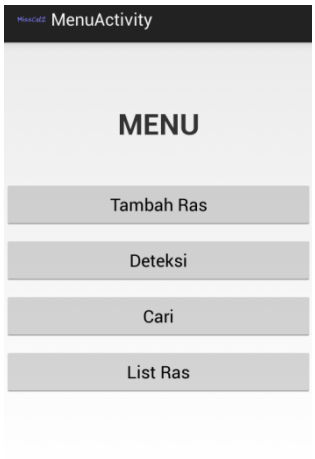
Tabel 4.1 Implementasi Antarmuka Aplikasi

No	Halaman	Keterangan	File XML
1	Menu	Halaman utama dari Aplikasi yang berisikan menu pilihan	Activity_menu
2	Tambah Ras	Halaman untuk menambah data ras kucing	Activity_tambah_ras
3	Deteksi	Halaman untuk melakukan *training* terhadap citra pelatihan dan pencarian ras kucing	Activity_deteksi
4	List Ras	Halaman untuk menampilkan jenis ras yang telah melakukan training	Activity_list_ras
5	Lihat Ras	Halaman untuk melihat data ras kucing	Activity_lihat_ras

Berikut adalah tampilan anta muka pengguna pada aplikasi perbandingan algoritma pengenalan wajah.

1. Halaman Menu

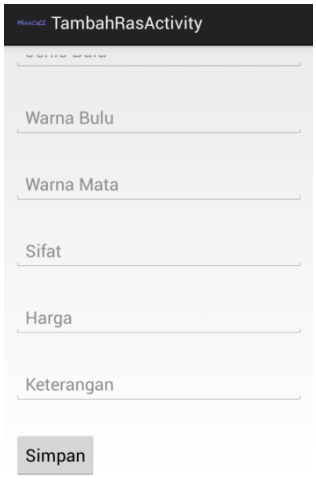
Halaman ini terdapat tiga pilihan tombol yaitu tombol tambah, tombol deteksi, tombol cari ras dan tombol liat ras.



Gambar 4.1 Halaman Menu

2. Halaman Tambah Ras

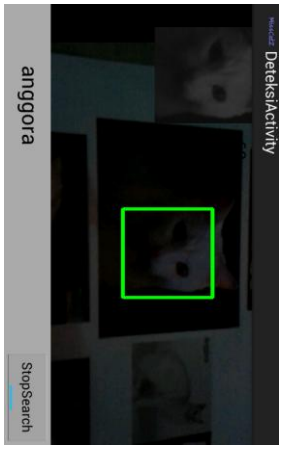
Halaman ini terdapat tiga pilihan tombol yaitu tombol tambah, tombol deteksi, tombol cari ras dan tombol liat ras.



Gambar 4.2 Halaman Tambah Ras

3. Halaman Deteksi

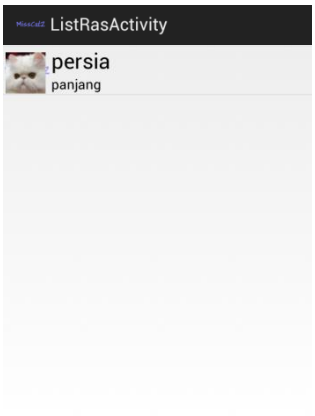
Halaman ini terdapat beberapa pilihan tombol yaitu tombol training dan tombol cari.



Gambar 4.3 Halaman Deteksi

4. Halaman List Ras

Halaman ini terdapat daftar dari ras kucing yang telah dilakukan training data.



Gambar 4.4 Halaman List Ras

5. Halaman Lihat Ras

Halaman ini terdapat data-data mengenai ras kucing yang telah dideteksi sebelumnya.



Gambar 4.5 Halaman Lihat Ras

4.1.4 Implementasi Kelas

Pada bagian implementasi kelas menggambarkan struktur file (.java) yang berisi kelas-kelas pada aplikasi pendeteksi ras. Adapun implementasi antarmuka yang digunakan untuk membangun aplikasi pendeteksi ras kucing adalah sebagai berikut:

Tabel 4.2 Implementasi Kelas

No	Kelas	Isi Kelas	File Java
1	Menu	onCreate onCreateOptionsMenu	MenuActivity
2	Tambah	onCreate onDestroy Simpan onCreateOptionMenu	TambahRasActivity
3	Deteksi	onCreate onManagerConnected onPause onResume onDestroy onCameraViewStarted onCameraViewStopped onTouch onCameraFrame onCreateOptionsMenu	DeteksiActivity
4	ListRas	onCreate onDestroy onCreateOptionsMenu onOptionsItemSelected	ListRasActivity
5	LihatRas	onCreate onDestroy	LihatRasActivity

		bacaData dariIdData onCreateOptionsMenu	
6	Kucing	Kucing	Kucing
7	SqliteManager	SqliteManager SqliteManagerHelper bukaKoneksi tutupKoneksi insertData updateData hapusData bacaData ambilData	SqliteManager

4.1.5 Implementasi Basis Data

Pembuatan basis data dilakukan dengan mengunakan aplikasi pemrograman SQLite. Implementasi basis data dalam bahasa SQLite dalam android adalah sebagai berikut :

1 Pembuatan Database

```
public SqliteManagerHelper(Context context) {

      super(context, NAMA_DATABASE, null, VERSI_DATABASE);

}
```

2 Pembuatan Tabel Training

```
public void onCreate(SQLiteDatabase database) {

      database.execSQL(

            "create table " + NAMA_TABEL + " (" +

            SqliteManager.FIELD_ID + " integer primary key
            autoincrement, " +

            SqliteManager.FIELD_RAS + " text not null, " +

            SqliteManager.FIELD_Bulu + " text not null," +

            SqliteManager.FIELD_Kepala + " text not null," +

            SqliteManager.FIELD_Telinga + "text not null," +

            SqliteManager.FIELD_Mata + " text not null," +

            SqliteManager.FIELD_Dagu + " text not null," +

            SqliteManager.FIELD_Hidung + " text not null," +

            SqliteManager.FIELD_Leher + " text not null," +
```

```
            SqliteManager.FIELD_Badan + " text not null," +
            SqliteManager.FIELD_Kaki + " text not null," +
            SqliteManager.FIELD_Ekor + " text not null," +
            SqliteManager.FIELD_Warna + " text not null," +
            SqliteManager.FIELD_Sifat + " text not null," +
            SqliteManager.FIELD_Harga + " text not null," +
            SqliteManager.FIELD_Ket + " text not null," +
            SqliteManager.FIELD_Path + " text not null " +
            ");";
      );
}
```

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk menguji hubungan antara program aplikasi yang dibuat dengan elemen yang lain dalam sistem. Adapun tujuan dari pengujian sistem ini adalah untuk memastikan semua elemen sistem sudah terhubung dengan baik. 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 jarak wajah terhadap webcam, dan pengaruh pencahayaan terhadap deteksi wajah. Pengujian pada *black box testing* yaitu menemukan kesalahan yang terdapat pada program.

4.2.1 Pengujian White Box

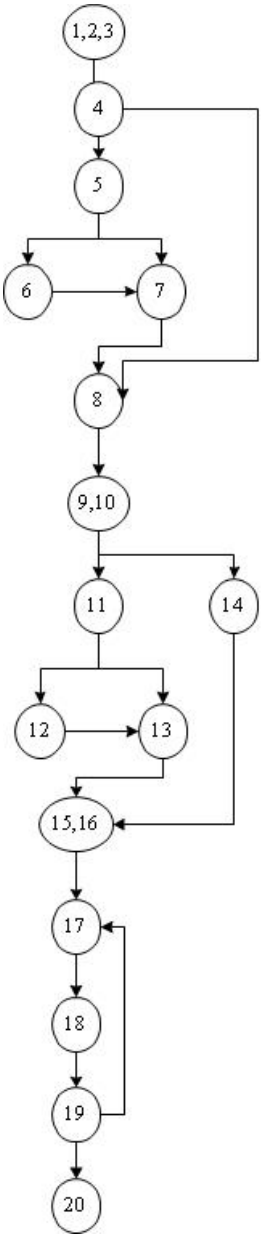
Pengujian white box digunakan untuk mengetahui kinerja logika dari algoritma yang telah dianalisis. Teknik yang digunakan yaitu Flow Graph yang digunakan untuk menggambarkan alur dari algoritma dan Graph Matriks yang digunakan untuk merepresentasikan flow graph dalam bentuk tabel.

Berikut ini adalah pengujian perangkat lunak pada metode pendeteksian image dengan metode viola-jones yang sebelumnya dilakukan pengenalan objek melalui OpenCV dengan function onCameraFrame. Pengujian ini dilakukan untuk pendeteksian wajah. Berikut ini adalah function yang akan diuji.

```
1   mRgba = inputFrame.rgba();
2   mGray = inputFrame.gray();
3   if (mAbsoluteFaceSize == 0)
4         int height = mGray.rows();
5         if (Math.round(height * mRelativeFaceSize) > 0)
6               mAbsoluteFaceSize = Math.round(height *
    mRelativeFaceSize);
7         endif
8   endif
9   MatOfRect faces = new MatOfRect();
10  if (mDetectorType == JAVA_DETECTOR)
11        if (mJavaDetector != null)
12              mJavaDetector.detectMultiScale(mGray, faces,
    1.1, 2, 2,
                 new Size(mAbsoluteFaceSize,
    mAbsoluteFaceSize), new Size());
13        Endif
    else
14        Log.e(TAG, "Detection method is not selected!");
15  Endif
16  Rect[] facesArray = faces.toArray();
17  for (int i = 0; i < facesArray.length; i++)
18        Core.rectangle(mRgba, facesArray[i].tl(),
    facesArray[i].br(), FACE_RECT_COLOR, 3);
19  Endfor
20  return mRgba;
```

a. Flow Graph metode pendeteksian wajah kucing

Berikut ini adalah flow graph dari pendeteksian :



Gambar 4.6 Flowgraph pendeteksian wajah kucing

Perhitungan white box :

1. Region = 6

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

2. Cyclomatic Complexity

V(G) = E – N + 2

= 20 – 16 + 2

= 6

3. Independent Path = 6

Path 1 = 1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20

Path 2 = 1,2,3,4,5,7,8,9,10,11,12,13,15,16,17,18,19,20

Path 3 = 1,2,3,4 ,8,9,10,11,12,13,15,16,17,18,19,20

Path 4 = 1,2,3,4,5, 6,7,8,9,10,14,15,16,17,18,19,20

Path 5 = 1,2,3,4,5,7,8,9,10,14,15,16,17,18,19,20

Path 6 = 1,2,3,4,8,9,10,14,15,16,17,18,19,20

Jumlah region, Cyclomatic Complexity dan Independent Path harus sama.

b. Graph matriks metode pendeteksian wajah

Satu node pada flow graph diwakilkan oleh satu nomor agar memudahkan perhitungan matriks. Tabel graph matriks dari metode pendeteksian ditunjukan dibawah ini

Tabel 4.3 Graph Matrix Pendeteksi Wajah

NODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Graph Matrix
1		1															0
2			1			1											1
3				1	1												1
4					1												0
5						1											0
6							1										0
7								1	1								1
8										1	1						1
9												1					0
10												1					0
11													1				0
12														1			0
3															1		0
														1		1	1
	Jumlah																5

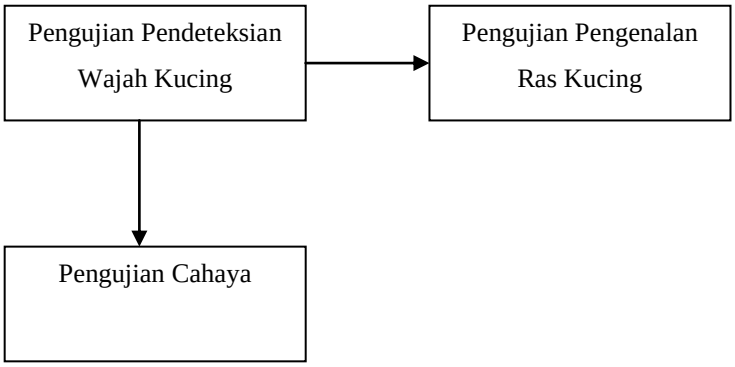
Keterangan :

Perhitungan nilai Cyclomatic Complexity = Nilai Graph Matrix + 1 = 5 + 1 = 6

4.2.2 Pengujian Performansi

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

Untuk menguji performansi, data yang telah di dapat dari pengujian pendeteksian wajah kucing digunakan kembali untuk pengujian-pengujian lainnya, untuk lebih jelasnya dapat dilihat pada skema pengujian performansi yaitu :



Gambar 4.7 Skema Pengujian Performansi

4.2.2.1 Pengujian Pendeteksian Wajah Kucing

Pengujian ini mengenai terdeteksi atau tidaknya wajah kucing yang telah diimplemntasikan dalam sisitem. Pendeteksian wajah kucing ini menggukan metode viola jones. Berikut adalah pengujian pendeteksian wajah kucing.

Tabel 4.4 Pengujian Pendeteksian Wajah Kucing

No	Jenis	Terdeteksi	Tidak Terdeteksi	Persentase Keberhasilan
1	Persia	15	15	50%
2	Exotic	10	20	33%
3	Angora	27	3	90%
4	Himalayan	2	28	6%
5	Domestik	16	14	53%

Rata-rata	70	80	46%

Dari tabel diatas dapat disimpulkan bahwa pendeteksian mengukan metode viola-jones dapat dilakukan dengan ras kucing yang paling sering terdeteksi adalah kucing anggora karena kucing angora murni hanya memiliki satu warna yaitu putih. Sedangkan kucing yang memiliki warna gelap atau bercorak sulit untuk terdeteksi seperti pada ras himalayan.

4.2.2.2 Pengujian Pengenalan Ras Kucing

Pengujian ini mengenai terdeteksi atau tidaknya ras kucing yang telah diimplemntasikan dalam sisitem setelah melakukan pendeteksian. Data dalam pengujian ini merupakan hasil dari pengujian sebelumnya yaitu pengujian pendeteksian wajah kucing dari data yang telah terdeteksi sebelumnya. Berikut adalah pengujian pendeteksian ras kucing.

Tabel 4.5 Pengujian Pengenalan Ras Kucing

No	Jenis	Dikenali	Tidak Dikenali	Persentase Keberhasilan
1	Persia	10	5	67%
2	Exotic	7	3	75%
3	Anggora	23	4	85%
4	Himalayan	1	1	50%
5	Domestik	11	5	68%
Rata - rata		52	18	75%

Dari tabel diatas dapat disimpulkan bahwa pengenalan ras menggunakan metode LBPH dapat dilakukan dengan rata-rata keberhasilan 75%.

4.2.2.3 Pengujian Cahaya

Pengujian ini tentang pengaruh cahaya terhadap proses deteksi wajah. Dalam pengujian pengaruh pencahayaan ini dilakukan di luar ataupun di dalam ruangan dalam keadaan terang, redup, dan gelap.

Tabel 4.6 Pengujian Cahaya

No	Jenis	Terang	Redup

1	Persia	√	√
2	Exotic	√	√
3	Anggora	√	√
4	Himalayan	√	√
5	Domestic	√	x

Dari tabel diatas dapat disimpulkan bahwa pencahayaan dengan intensitas terang menampilkan hasil maksimal dalam perdeteksian wajah kucing. Tetapi cocok pula untuk wilayah dengan intensitas cahaya redup.

Keterangan

√ : terdeteksi

x : tidak terdeteksi

4.2.3 Pengujian Black Box

Pengujian *blackbox* terfokus pada menemukan kesalahan yang terdapat pada program. Sebelum melakukan pengujian dilakukan dahulu rencana pengujian. Rencana pengujian yang akan dilakukan dapat dilihat sebagai berikut:

Tabel 4.7 Rencana Pengujian

No	Komponen yang diuji	Skenario pengujian	Jenis pengujian
1	Kamera	Memberikan akses kepada kamera	*Black box*
		Capture video secara realtime	*Black box*
2	Interaksi pengguna	Memilih tombol pada menu	*Black box*
		Menekan tombol capture image	*Black box*
		Menekan tombol simpan ras	*Black box*
		Menekan tombol edit ras	*Black box*
		Menekan tombol hapus ras	*Black box*
		Menekan tombol kembali	*Black box*
3	Deteksi	Mendeteksi wajah kucing	*Black box*
		Mendeteksi ras kucing	*Black box*
4	Database	Menyimpan data di database	*Black box*
		Mengubah data dari database	*Black box*
		Menghapus data dari database	*Black box*
		Melihat data dari database	*Black box*

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 Pengujian

No	Pengujian	Skenario pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Kamera	Memberikan akses kepada kamera	Kamera bekerja setelah pengguna memilih tombol deteksi	[√]Berhasil [ ]Tidak Berhasil
		Capture video secara realtime	Kamera mencapture setelah pengguna memilih tombol rec	[√]Berhasil [ ]Tidak Berhasil
2	Interaksi pengguna	Memilih tombol pada menu	Menampilkan form sesuai menu pilihan pengguna	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol capture	Menyimpan hasil capture oleh kamera	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol simpan	Menyimpan gambar hasil capture kamera	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol edit ras	Menampilkan form edit data ras	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol hapus ras	Menampilkan alert penghapusan data ras	[√]Berhasil [ ]Tidak Berhasil
3	Deteksi	Mendeteksi wajah kucing	Sistem dapat mendeteksi wajah pengguna melalui kamera	[√]Berhasil [ ]Tidak Berhasil
		Mendeteksi ras kucing	Sistem dapat mendeteksi ras	[√]Berhasil [ ]Tidak Berhasil
4	Database	Menyimpan data	Berhasil menyimpan	[√]Berhasil

		di database	data ke database	[ ]Tidak Berhasil
		Mengubah data dari database	Berhasil mengubah data dari database	[√]Berhasil [ ]Tidak Berhasil
		Menghapus data dari database	Berhasil menghapus data dari database	[√]Berhasil [ ]Tidak Berhasil
		Melihat data dari database	Berhasil menampilkan data dari database	[√]Berhasil [ ]Tidak Berhasil

4.3 Kesimpulan Pengujian

Berdasarkan hasil pengujian pengujian performansi dan *black box testing* dengan kasus uji di atas dapat ditarik kesimpulan bahwa pembangunan aplikasi pendeteksian ras kucing tidak terdapat kesalahan proses dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan, tetapi terdapat pula kekurangan seperti tidak dapat mendeteksi kucing yang berwarna gelap dan bercorak pada wajahnya sehingga diharapkan adanya penelitian lebih lanjut baik oleh peneliti maupun pihak lainnya untuk menyempurnakan penelitian ini.

{Halaman ini dikosongkan}