BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Pada bab ini akan dilakukan implementasi dan pengujian terhadap perangkat lunak aplikasi identifikasi ruas jari tangan. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan kedalam bahasa pemograman. Setelah implementasi maka dilakukan pengujian dan akan dilihat kekurangan-kekurangan untuk pengembangan sistem selanjutnya.

Setelah aplikasi dianalisis dan dirancang secara rinci, tahap selanjutnya adalah implementasi. Tujuan dari implementasi adalah untuk mengkonfirmasi modul-modul perancangan sehingga pengguna dapat memberikan masukan kepada pengembangan sistem.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan sistem dapat dilihat pada Tabel 4.1.

Tabel 4.1 Spesifikasi perangkat keras

No	Perangkat Keras	Spesifikasi
1	Processor	Intel Core i3 2.30 GHz
2	Monitor	Monitor 14inch 1366 x 768 (32bit) (60Hz)
3	VGA	512 MB
4	Memori	2 GB
5	Kamera	2.0 Megapixel
6	Keyboard dan Mouse	

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun dan mengimplementasikan sistem dapat dilihat pada Tabel 4.2.

Tabel 4.2 Spesifikasi perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7
2	Tools	Microsoft Visual Studio 2010
3	Bahasa Pemrograman	C#

4.1.3 Implementasi Basis Data

Pembuatan basis data atau *database* menggunakan bahasa *SQL*, implementasinya sebagai berikut :

1. Tabel Gambar_Latih

Tabel 4.3 SQL tabel gambar_latih

```
CREATE TABLE IF NOT EXISTS `Gambar_Latih` (
  `id_gambarlatih` char(5) NOT NULL,
  `id_gambarasli` char(5) NOT NULL,
  `nama_pemilik` varchar (30) NOT NULL,
  `gambar_samplelatih ` longblob NOT NULL,
  PRIMARY KEY (`id_gambarlatih`),
  KEY `id_gambarasli` (`id_gambarasli `),
  KEY `nama_pemilik` (`nama_pemilik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `Gambar_Latih`
  ADD CONSTRAINT `Gambar_Latih_fk_1` FOREIGN KEY
(`id_gambarasli`) REFERENCES `Gambar` (`id_gambarasli`)
ON UPDATE CASCADE,
  ADD CONSTRAINT `Gambar_Latih_fk_2` FOREIGN KEY
(`nama_pemilik`) REFERENCES `Gambar` (`id_gambarasli`) ON
UPDATE CASCADE;
```

2. Tabel Gambar

Tabel 4.4 SQL tabel gambar

```
CREATE TABLE IF NOT EXISTS `Gambar` (
  `id_gambarasli` char(5) NOT NULL,
  `nama_pemilik` varchar (30) NOT NULL,
  `gambar_sample1 ` longblob NOT NULL,
  `gambar_sample2 ` longblob NOT NULL,
```

```
   `gambar_sample3 ` longblob NOT NULL,
   `gambar_sample4 ` longblob NOT NULL,
   PRIMARY KEY (`id_gambarasli`),
   KEY `nama_pemilik` (`nama_pemilik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `Gambar`
  ADD     CONSTRAINT     `Gambar_fk_1`     FOREIGN     KEY
(`nama_pemilik`)  REFERENCES  `Pemilik`  (`id_pemilik`)  ON
UPDATE CASCADE;
```

3. Tabel Pemilik

Tabel 4.5 SQL tabel pemilik

```
CREATE TABLE IF NOT EXISTS `Pemilik` (
   `id_pemilik` char(5) NOT NULL,
   `nama_pemilik` varchar (30) NOT NULL,
   PRIMARY KEY (`id_pemilik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. Tabel Gambar_Uji

Tabel 4.6 SQL tabel gambar_uji

```
CREATE TABLE IF NOT EXISTS `Gambar_Uji` (
   `id_gambaruji` char(5) NOT NULL,
   `id_gambarasli` char(5) NOT NULL,
   `nama_pemilik` varchar (30) NOT NULL,
   `gambar_sampleuji` longblob NOT NULL,
   PRIMARY KEY (`id_gambaruji`),
   KEY `id_gambarasli` (`id_gambarasli `),
   KEY `nama_pemilik` (`nama_pemilik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `Gambar_Uji`
  ADD    CONSTRAINT    `Gambar_Uji_fk_1`    FOREIGN    KEY
(`id_gambarasli`)  REFERENCES  `Gambar`  (`id_gambarasli`)
ON UPDATE CASCADE,
   ADD    CONSTRAINT    `Gambar_Uji_fk_2`    FOREIGN    KEY
(`nama_pemilik`) REFERENCES `Gambar` (`id_gambarasli`) ON
UPDATE CASCADE;
```

5. Tabel Processing_Latih

Tabel 4.7 SQL tabel processing_latih

```
CREATE TABLE IF NOT EXISTS `Processing_Latih` (
   `id_processinglatih` integer UNSIGNED NOT NULL
AUTO_INCREMENT,
   `id_gambarlatih` char(5) NOT NULL,
   `nilai_grayscale` double NOT NULL,
   `nilai_ekstraksiLBP` double NOT NULL,
   `nilai_vektorcirilatih` double NOT NULL,
   PRIMARY KEY (`id_processinglatih`),
```

```
    KEY `id_gambarlatih` (`id_gambarlatih`),
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;


 ALTER TABLE `Processing_Latih`
   ADD CONSTRAINT `Processing_Latih_fk_1` FOREIGN KEY
 (`id_gambarlatih`)       REFERENCES       `Gambar_Latih`
 (`id_gambarlatih`) ON UPDATE CASCADE;
```

6. Tabel Processing_Uji

Tabel 4.8 SQL tabel processing_uji

```
  CREATE TABLE IF NOT EXISTS `Processing_Uji` (
    `id_processinguji` integer UNSIGNED NOT NULL
  AUTO_INCREMENT,
    `id_gambaruji` char(5) NOT NULL,
    `nilai_grayscale` double NOT NULL,
    `nilai_ekstraksiLBP` double NOT NULL,
    `nilai_vektorciriuji` double NOT NULL,
    PRIMARY KEY (`id_processinguji`),
    KEY `id_gambaruji` (`id_gambaruji`),
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;


 ALTER TABLE `Processing_Uji`
    ADD CONSTRAINT `Processing_Uji_fk_1` FOREIGN KEY
  (`id_gambaruji`)         REFERENCES         `Gambar_Uji`
  (`id_gambaruji`) ON UPDATE CASCADE;
```

7. Tabel Penyimpanan_Data

Tabel 4.9 SQL tabel penyimpanan_data

```
  CREATE TABLE IF NOT EXISTS `Penyimpanan_Data` (
    `id_penyimpanandata` integer UNSIGNED NOT NULL
  AUTO_INCREMENT,
    `id_processinglatih` integer NOT NULL,
    `id_processinguji` integer NOT NULL,
    `nilai_vektorcirilatih` double NOT NULL,
    `nilai_vektorciriuji` double NOT NULL,
    PRIMARY KEY (`id_penyimpanandata`),
    KEY `id_processinglatih` (`id_processinglatih`),
    KEY `id_processinguji` (`id_processinguji`)
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;


 ALTER TABLE `Penyimpanan_Data`
   ADD CONSTRAINT `Penyimpanan_Data_fk_1` FOREIGN KEY
 (`id_processinglatih`)    REFERENCES    `Processing_Latih`
 (`id_processinglatih`) ON UPDATE CASCADE,
    ADD CONSTRAINT `Penyimpanan_Data_2` FOREIGN KEY
  (`id_processinguji`)     REFERENCES     `Processing_Uji`
  (`id_processinguji`) ON UPDATE CASCADE;
```

8. Tabel Identifikasi

Tabel 4.10 SQL tabel identifikasi

```
CREATE TABLE IF NOT EXISTS `Identifikasi` (
  `id_identifikasi` integer UNSIGNED NOT NULL
AUTO_INCREMENT,
  `id_penyimpanandata` integer NOT NULL,
  `id_gambarasli` char(5) NOT NULL,
  `nilai_vektorcirilatih` double NOT NULL,
  `nilai_vektorciriuji` double NOT NULL,
  `nilai_euclidean_distance` double NOT NULL,
  `nama_pemilik ` varchar(30) NOT NULL,
  PRIMARY KEY (`id_identifikasi`),
  KEY `id_penyimpanandata` (`id_penyimpanandata`),
  KEY `id_gambarasli` (`id_gambarasli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


ALTER TABLE `Identifikasi`
  ADD CONSTRAINT `Identifikasi_fk_1` FOREIGN KEY
(`id_penyimpanandata`) REFERENCES `Penyimpanan_Data`
(`id_penyimpanandata`) ON UPDATE CASCADE,
  ADD CONSTRAINT `Identifikasi_fk_2` FOREIGN KEY
(`id_gambarasli`) REFERENCES `Gambar` (`id_gambarasli`)
ON UPDATE CASCADE;
```

4.1.4 Implementasi Antarmuka Sistem

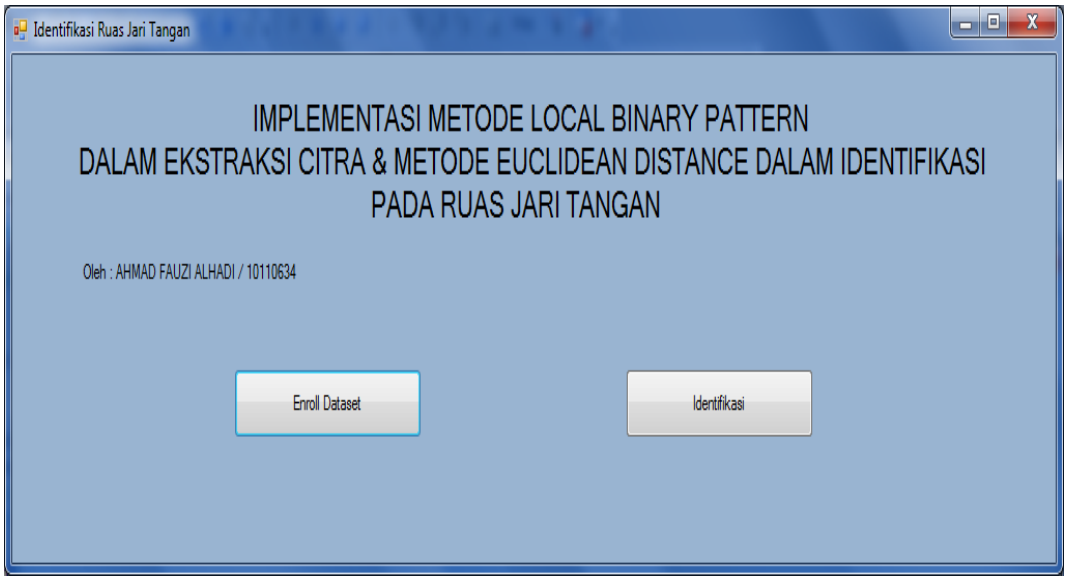
Dari perancangan antarmuka yang telah dibuat pada bab sebelumnya, maka tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan. Implementasi antarmuka sistem meliputi :

Tabel 4.11 Implementasi antarmuka sistem

No	Nama Menu	Nama File	Keterangan
1	Form Menu Utama	MenuUtama.cs	Tampilan utama aplikasi
2	Form Proses Latih Citra	ProsesLatih.cs	Tampilan pada proses latih citra
3	Form Proses Uji Citra & Identifikasi	ProsesUjiIdentifikasi.cs	Tampilan pada proses uji citra & identifikasi

4.1.4.1 Tampilan Menu Utama

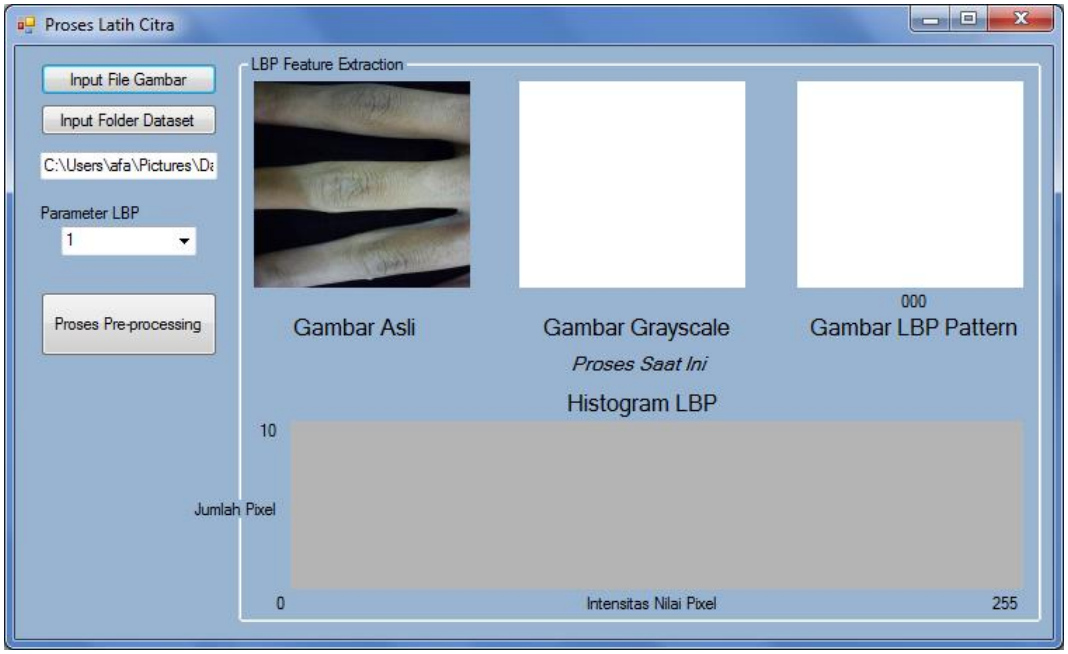
Tampilan menu utama merupakan tampilan awal yang pertama kali muncul ketika aplikasi dijalankan. Tampilan ini merupakan tahapan pemilihan proses identifikasi ruas jari tangan. Tampilan menu utama aplikasi dapat dilihat pada Gambar 4.1 .



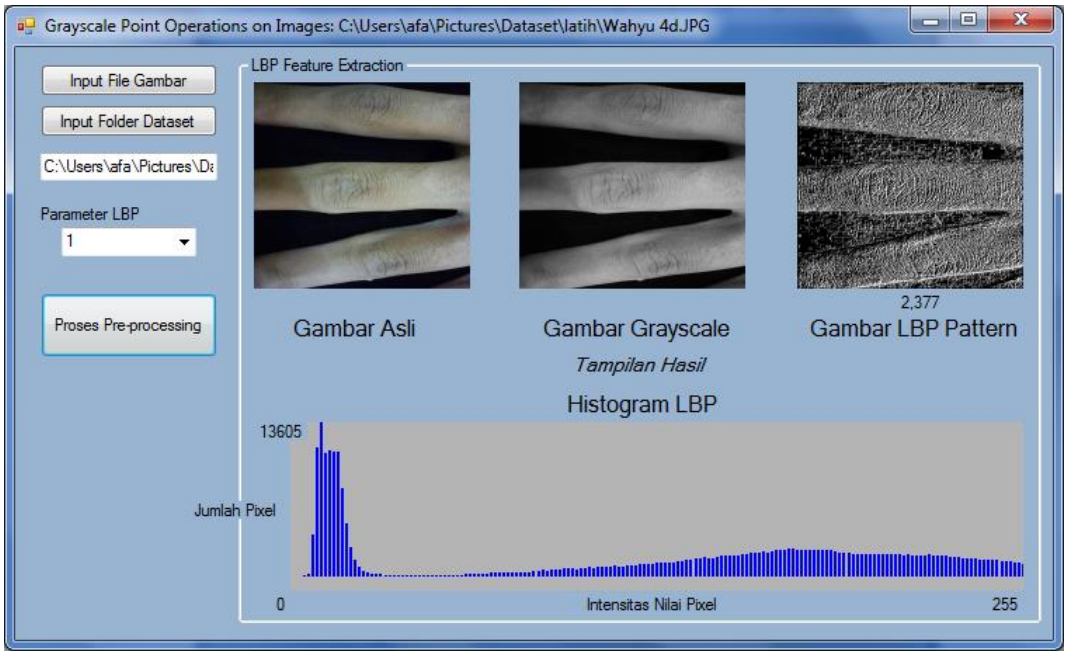
Gambar 4.1 Tampilan menu utama

4.1.4.2 Tampilan Form Proses Latih Citra

Tampilan ini muncul ketika memilih button Enroll Dataset pada menu utama, yaitu tampilan form proses latih citra yang digunakan untuk proses pemasukan dataset gambar latih dan didalamnya terdapat 3 button, yaitu Input File Gambar, Input Folder Dataset dan Proses Pre-processing. Tampilan dapat dilihat pada gambar Gambar 4.2 & Gambar 4.3 .



Gambar 4.2 Tampilan form proses latih citra setelah memasukan gambar



Gambar 4.3 Tampilan form proses latih citra hasil *pre-processing*

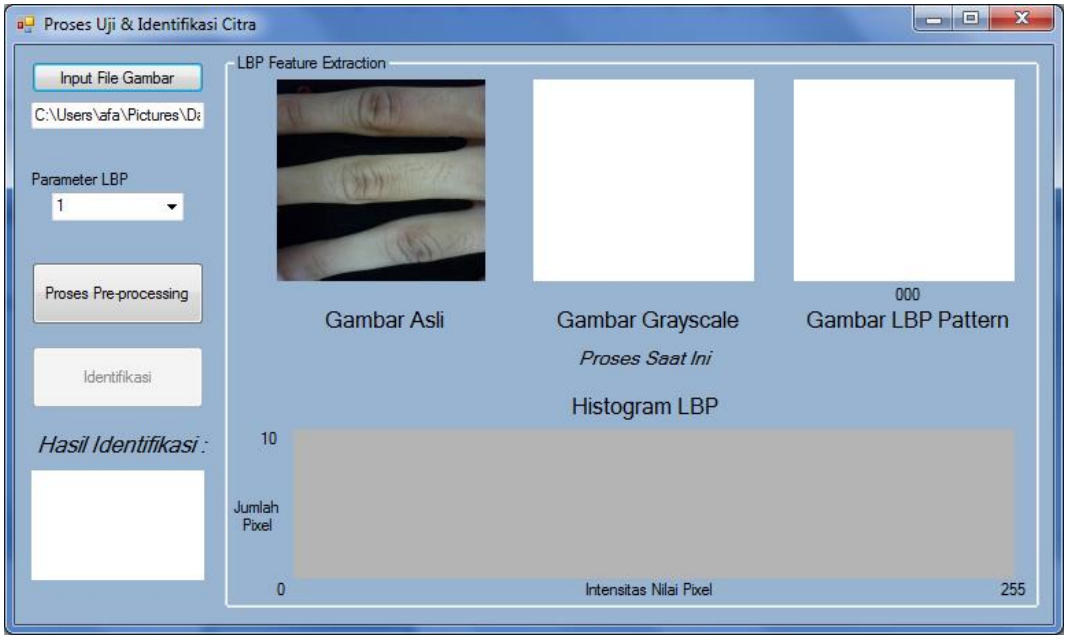
Penjelasan dari 3 button tersebut :

a. Input File Gambar : Memasukan citra berupa 1 file gambar sebagai citra target.

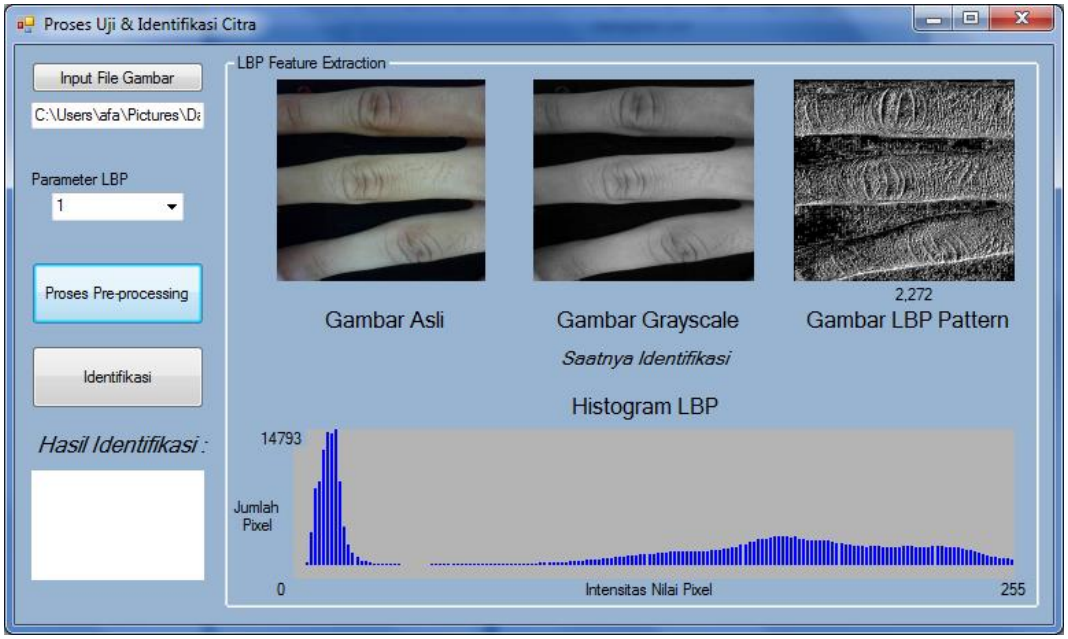
b. Input Folder Dataset : Memasukan citra berupa folder dataset sebagai citra target.
c. Proses Pre-processing : Melakukan proses *pre-processing* gambar citra.

4.1.4.3 Tampilan Menu Proses Uji & Identifikasi Citra

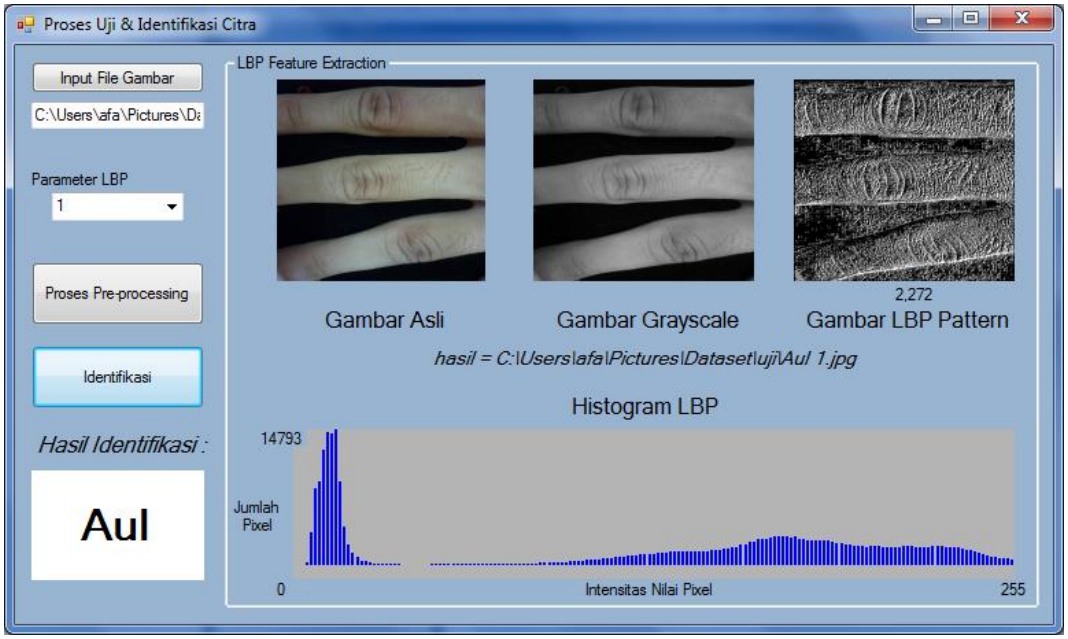
Tampilan ini muncul ketika memilih button Identifikasi pada menu utama, yaitu tampilan form proses uji & identifikasi citra yang digunakan untuk proses uji gambar yang akan diidentifikasi dan didalamnya terdapat 3 button, yaitu Input File Gambar, Proses Pre-processing dan Identifikasi. Tampilan dapat dilihat pada Gambar 4.4 , Gambar 4.5 & Gambar 4.6



Gambar 4.4 Tampilan form proses setelah memasukan gambar uji



Gambar 4.5 Tampilan form proses uji & identifikasi citra hasil *pre-processing*



Gambar 4.6 Tampilan form proses setelah proses hasil identifikasi

4.1.4.4 Tampilan Pesan Jika Membatalkan Proses Input Gambar

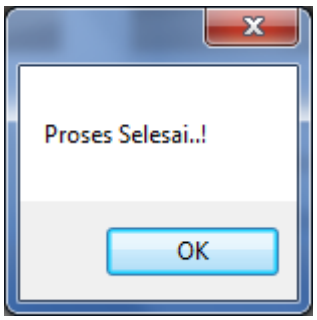
Tampilan pesan ini muncul apabila pengguna membatalkan proses input file gambar & input folder dataset untuk memasukan gambar citra target. Tampilan pesan pemberitahuan apabila membatalkan proses input gambar citra target dapat dilihat pada Gambar 4.7 .



Gambar 4.7 Tampilan pesan jika membatalkan proses input gambar

4.1.4.5 Tampilan Pesan Jika Proses *Pre-processing* Gambar Selesai

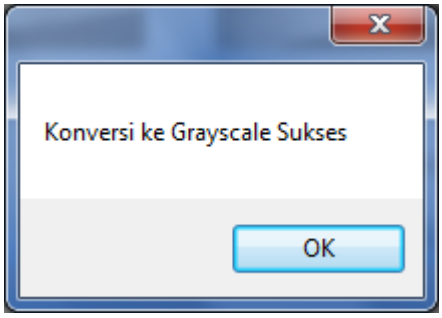
Tampilan pesan ini muncul apabila proses *pre-processing* gambar telah selesai. Tampilan pesan pemberitahuan apabila proses *pre-processing* gambar telah selesai dapat dilihat pada Gambar 4.8 .



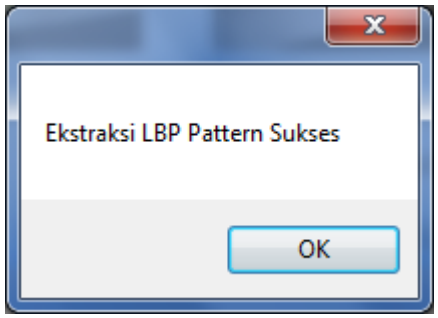
Gambar 4.8 Tampilan pesan jika proses *pre-processing* gambar selesai

4.1.4.6 Tampilan Pesan Jika Proses *Grayscale* dan Ekstraksi Sukses

Tampilan pesan ini muncul apabila proses *grayscale* dan ekstraksi citra gambar sukses. Tampilan pesan pemberitahuan apabila proses *grayscale* dan ekstraksi citra gambar sukses dapat dilihat pada Gambar 4.9 dan Gambar 4.10 .



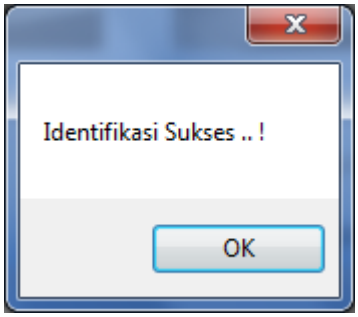
Gambar 4.9 Tampilan pesan jika proses g*rayscale* sukses



Gambar 4.10 Tampilan pesan jika proses ekstraksi sukses

4.1.4.7 Tampilan Pesan Jika Proses Identifikasi Sukses

Tampilan pesan ini muncul apabila proses identifikasi berjalan sukses. Tampilan pesan pemberitahuan apabila proses identifikasi berjalan sukses dapat dilihat pada gambar Gambar 4.11 .



Gambar 4.11 Tampilan pesan jika proses identifikasi sukses

4.2 Pengujian Sistem

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak (*software*). Tujuan dari pengujian adalah untuk menemukan kesalahan dan kekurangan pada perangkat lunak. Pengujian juga bermaksud untuk menjamin perangkat lunak yang di bangun memilki kualitas yang handal, yaitu mampu melakukan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

4.2.1 Proses Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui apakah rancangan dan implementasi yang sudah dilakukan berjalan dengan prosedur yang diinginkan atau tidak. Pengujian ini dilakukan dengan menggunakan parameter pengujian seperti mengukur akurasi hasil pendeteksian dan waktu proses komputasi berdasarkan parameter LBP.

4.2.2 Skenario Pengujian

Skenario pengujian aplikasi identifikasi ruas jari tangan ini menggunakan metode pengujian *white box* dan *black box*. Pengujian menggunakan metode pengujian *white box* berfokus pada pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural, sedangkan pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Skenario pengujian dapat dilihat pada Tabel 4.12 yang akan dijalankan pada sistem ini.

Tabel 4.12 Skenario pengujian perangkat lunak

Pengujian	Butir Uji	Jenis Pengujian
Metode *Local Binary Pattern*	Algoritma LBP dalam ekstraksi ciri citra	*White Box*
Metode *Euclidean Distance*	Algoritma *euclidean distance* dalam identifikasi	*White Box*
Input File Gambar	Proses pemasukan file gambar latih dan uji yang ingin diidentifikasi.	*Black Box*
Input Folder Dataset	Proses pemasukan file folder gambar latih yang ingin diidentifikasi.	*Black Box*
Proses *Pre-processing*	Proses waktu komputasi yang dilakukan.	*Black Box*
Identifikasi	Proses pendeteksian gambar latih dan uji	*Black Box*
	Persenan keakuratan hasil identifikasi	*Black Box*

4.2.3 Pengujian *White Box*

Bedasarkan skenario pengujian *white box* yang telah disusun pada Tabel 4.12, maka dapat dilakukan pengujian sebagai berikut :

4.2.3.1 Pengujian Algoritma *Local Binary Pattern* pada Esktraksi Ciri Citra

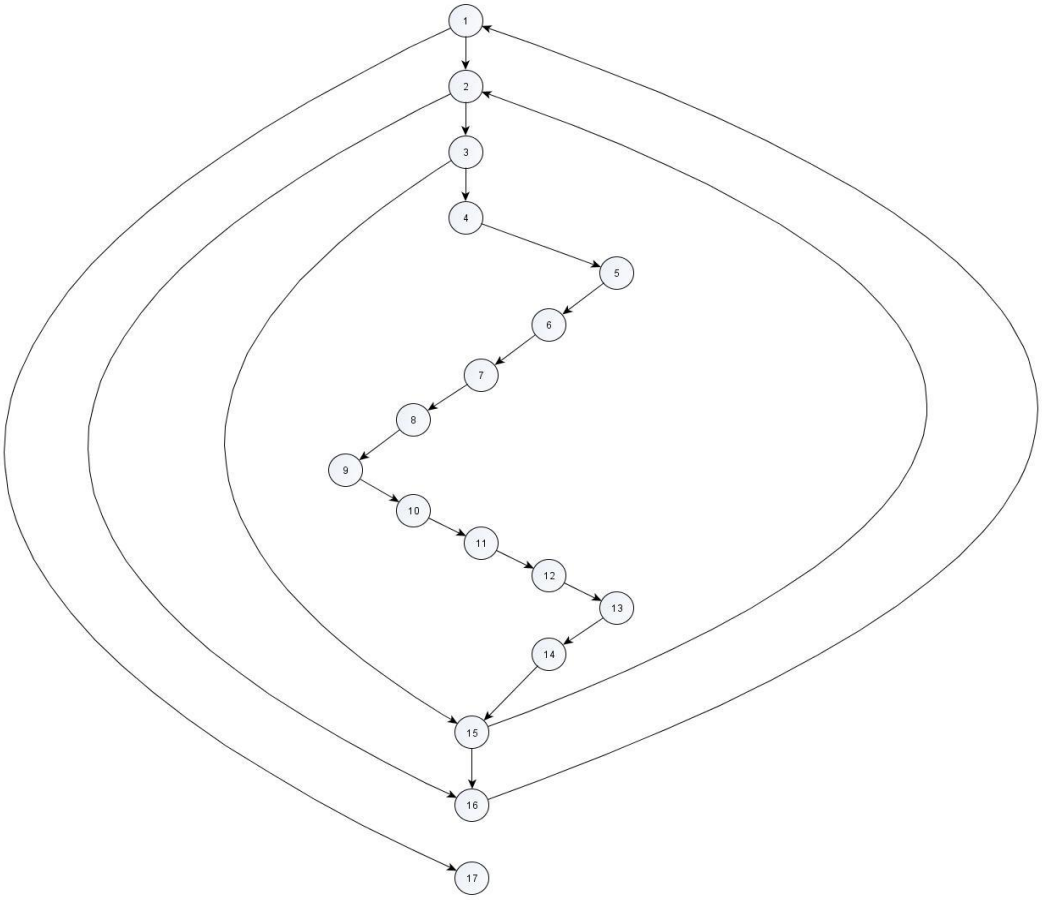
Berikut ini adalah pengujian algoritma *Local Binary Pattern*, pengujiannya dapat dilihat dibawah ini :

```
1: {
      dst = Mat::zeros(src.rows-2, src.cols-2, CV_8UC1);
2:      for(int i=1;i<src.rows-1;i++)
     {
3:        for(int j=1;j<src.cols-1;j++)
        {
4:             _Tp center = src.at<_Tp>(i,j);
5:             unsigned char code = 0;
6:             code |= (src.at<_Tp>(i-1,j-1) > center) << 7;
7:             code |= (src.at<_Tp>(i-1,j) > center) << 6;
8:             code |= (src.at<_Tp>(i-1,j+1) > center) << 5;
```

```
9:            code |= (src.at<_Tp>(i,j+1) > center) << 4;
10:           code |= (src.at<_Tp>(i+1,j+1) > center) << 3;
11:           code |= (src.at<_Tp>(i+1,j) > center) << 2;
12:           code |= (src.at<_Tp>(i+1,j-1) > center) << 1;
13:           code |= (src.at<_Tp>(i,j-1) > center) << 0;
14:           dst.at<unsigned char>(i-1,j-1) = code;
15:       }
16:     }
17: }
```

Gambar 4.12 *Pseudecode* algoritma LBP

1. *Flow Graph* Algoritma *Local Binary Pattern* pada Esktraksi Ciri Citra.



Gambar 4.13 *Flow Graph* LBP

Keterangan Gambar 4.13 :

◯ : Menggambarkan kondisi

⟶ : Menggambarkan aksi

Dari Gambar 4.13 dapat dihitung *cyclomatic complexity* sebagai berikut :

V(G) = E – N + 2

V(G) = 20 – 17 + 2

V(G) = 5

Dimana :

E = Jumlah *edge* pada grafik alir

N = Jumlah *node* pada grafik alir

Jadi, *cyclomatic complexity* untuk Gambar 4.13 adalah 5. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 5 *path* yang terdiri dari :

Path 1 : 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
Path 2 : 1,2,3,4,15,16,17
Path 3 : 1,2,3,15,16,17
Path 4 : 1,2,16,17
Path 5 : 1,17

2. *Graph Matriks* Algoritma *Local Binary Pattern* pada Esktraksi Ciri Citra.

Tabel 4.13 *Graph Matriks* algoritma LBP

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	
1		1															1	1
2			1													1		1
3				1											1			1
4					1													0
5						1												0
6							1											0
7								1										0
8									1									0
9										1								0
10											1							0
11												1						0
12													1					0
13														1				0
14															1			0

15		1														1		1
16	1																	0
17																		0
																Jumlah +1		**5**

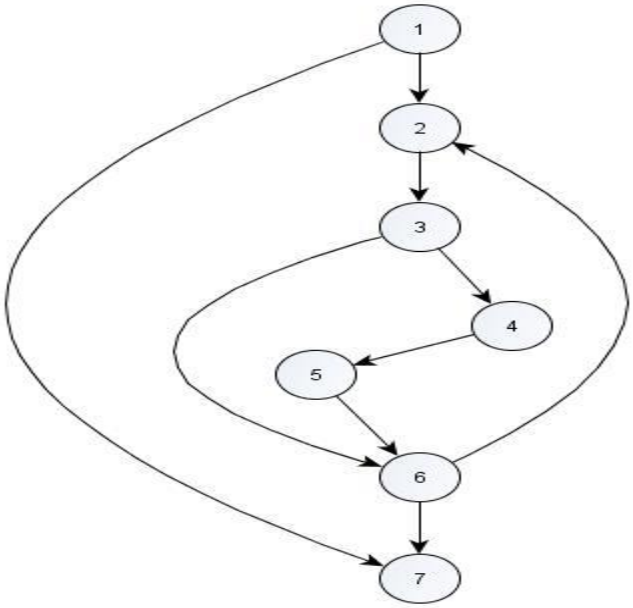
4.2.3.2 Pengujian Algoritma *Euclidean Distance* pada Identifikasi

Berikut ini adalah pengujian algoritma *Euclidean Distance*, pengujiannya dapat dilihat dibawah ini :

```
1: function EuclidenaDistance
2:         dist=0
3:         for d=1 to N // d = dimension
4:               dist=dist+(x1[d]-x2[d]^2
5:         next
6:         return sqrt(dist)
7: end function
```

Gambar 4.14 *Pseudecode* algoritma *Euclidean Distance*

1. *Flow Graph* Algoritma *Euclidean Distance* pada Identifikasi.



Gambar 4.15 *Flow Graph Euclidean Distance*

Keterangan Gambar 4.15 :

 : Menggambarkan kondisi

 : Menggambarkan aksi

Dari Gambar 4.15 dapat dihitung *cyclomatic complexity* sebagai berikut :

V(G) = E – N + 2

V(G) = 9 – 7 + 2

V(G) = 4

Dimana :

E = Jumlah *edge* pada grafik alir

N = Jumlah *node* pada grafik alir

Jadi, *cyclomatic complexity* untuk Gambar 4.15 adalah 4. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 4 *path* yang terdiri dari :

Path 1 : 1,2,3,4,5,6,7
Path 2 : 1,2,3,4,6,7
Path 3 : 1,2,3,6,7
Path 4 : 1,2,7

2. *Graph Matriks* Algoritma *Euclidean Distance* pada Identifikasi.

Tabel 4.14 *Graph Matriks* algoritma *Euclidean Distance*

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
1		1					1	1
2			1					0
3				1		1		1
4					1			0
5						1		0
6		1					1	1
7								0
							Jumlah +1	**4**

4.2.4 Kasus dan Hasil Pengujian *Black Box*

Bedasarkan skenario pengujian *black box* yang telah disusun pada Tabel 4.12, maka dapat dilakukan pengujian sebagai berikut :

4.2.4.1 Skenario Pengujian Input File Gambar

Berikut adalah pengujian yang dilakukan untuk mengetahui apakah fungsionalitas input file gambar pada aplikasi dapat berjalan sesuai yang diharapkan atau tidak. Pengujian ini dapat dilihat pada Tabel 4.15 .

Tabel 4.15 Skenario pengujian input file gambar

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Gambar sample latih dan uji citra. - Resolusi 480x640 - 8 bit	Dapat menampilkan *open file dialog* untuk memasukan gambar latih & uji target sasaran dan gambar muncul pada aplikasi.	Proses dapat menampilkan *open file dialog* dan memunculkan gambar pada aplikasi.	Berhasil
Gambar bukan sample latih & uji citra	Dapat menampilkan *open file dialog* untuk memasukan gambar ke aplikasi tetapi gambar tidak dapat masuk ke aplikasi.	Proses dapat menampilkan *open file dialog* dan gambar tidak dapat masuk ke aplikasi	Berhasil

4.2.4.2 Skenario Pengujian Input Folder Dataset

Berikut adalah pengujian yang dilakukan untuk mengetahui apakah fungsionalitas input folder dataset pada aplikasi dapat berjalan sesuai yang diharapkan atau tidak. Pengujian ini dapat dilihat pada Tabel 4.16 .

Tabel 4.16 Skenario pengujian input folder dataset

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Folder berupa gambar sample latih citra. - Resolusi 480x640 - 8 bit	Dapat menampilkan *open file dialog* untuk memasukan folder gambar latih target sasaran dan gambar muncul pada aplikasi.	Proses dapat menampilkan *open file dialog* dan memunculkan gambar pada aplikasi.	Berhasil
Folder berupa gambar bukan sample latih citra.	Dapat menampilkan *open file dialog* untuk memasukan folder gambar ke aplikasi tetapi gambar tidak dapat masuk ke aplikasi.	Proses dapat menampilkan *open file dialog* dan gambar tidak dapat masuk ke aplikasi	Berhasil

4.2.4.3 Skenario Pengujian Proses *Pre-processing*

Berikut adalah pengujian yang dilakukan untuk mengetahui apakah fungsionalitas proses *pre-processing* gambar pada aplikasi dapat berjalan sesuai yang diharapkan atau tidak. Pengujian ini dapat dilihat pada Tabel 4.17 .

Tabel 4.17 Skenario pengujian proses *pre-processing*

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Gambar asli latih dan uji citra yang sudah diinput	Dapat melakukan proses *pre-processing* dari tahap *grayscale* sampai ekstraksi ciri citra dan menampilkan hasil *pre-processing*.	Proses dapat melakukan *pre-processing* dan menampilkan gambar hasilnya.	Berhasil

4.2.4.4 Skenario Pengujian Identifikasi

Berikut adalah pengujian yang dilakukan untuk mengetahui apakah fungsionalitas identifikasi gambar latih & uji pada aplikasi dapat berjalan sesuai yang diharapkan atau tidak. Pengujian ini dapat dilihat pada Tabel 4.18 .

Tabel 4.18 Skenario pengujian identifikasi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nilai fitur vektor ciri latih dan uji dari penyimpanan pengolahan data sistem.	Dapat melakukan proses pengenalan atau identifikasi gambar uji dengan gambar latih dan menampilkan hasil identifikasi.	Proses dapat melakukan pengidentifikasian atau pengenalan gambar dan menampilkan hasil identifikasi.	Berhasil
Nilai fitur vektor ciri uji, tetapi tidak memiliki fitur vektor ciri data latih.	Tidak dapat melakukan proses identifikasi gambar dan menampilkan proses identifikasi gagal.	Proses tidak dapat diidentifikasi dan menampilkan pesan proses gagal.	Berhasil

4.2.4.5 Kesimpulan Pengujian *Black Box*

Berdasarkan hasil pengujian *black box* yang telah dilakukan dapat diambil kesimpulan bahwa pada proses masih memungkinkan terjadi kesalahan tetapi secara fungsional sistem sudah dapat menghasilkan output yang diharapkan.

4.2.5 Kasus dan Hasil Pengujian Aplikasi

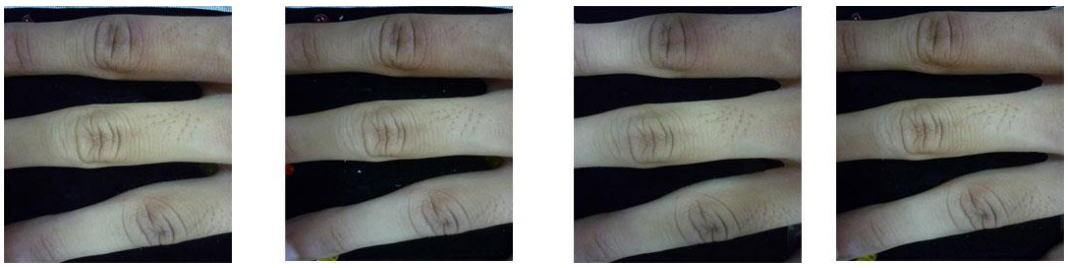
Berikut merupakan pengujian yang dilakukakn terhadap gambar citra ruas jari tangan dari berbagai orang dan *sample* yang berbeda-beda. Pengujian yang dilakukan untuk melihat keakuratan atau keberhasilan dalam mengenali sebuah gambar citra yang akan dikenali. Berikut gambar-gambar file dataset yang akan diidentifikasi :



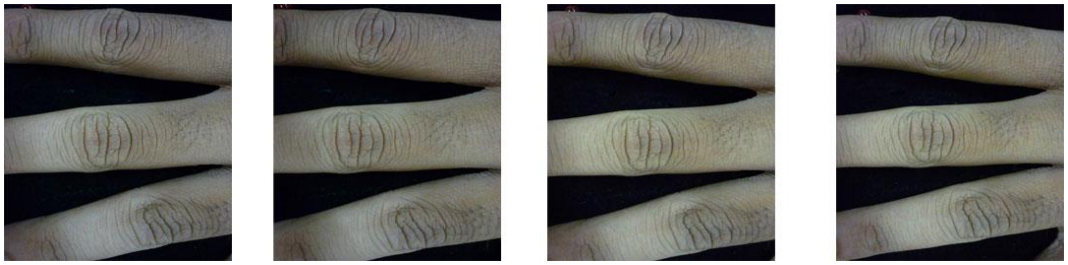
Gambar 4.16 Ruas jari tangan Adil (sample 1-4)



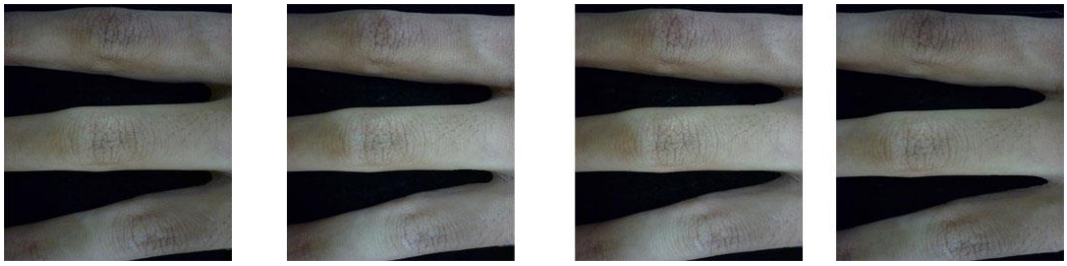
Gambar 4.17 Ruas jari tangan Adli (sample 1-4)



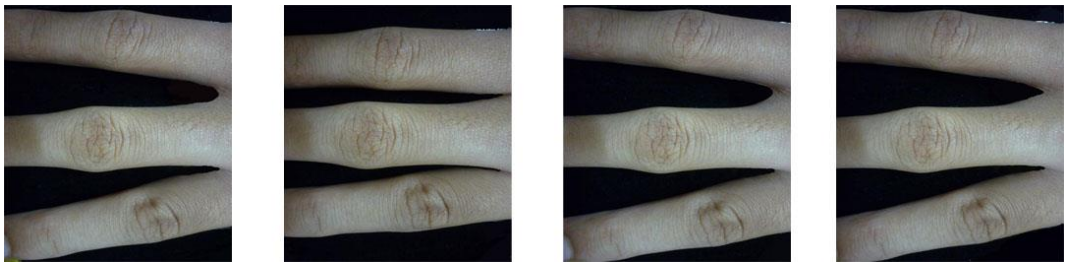
Gambar 4.18 Ruas jari tangan Aul (sample 1-4)



Gambar 4.19 Ruas jari tangan Authar (sample 1-4)



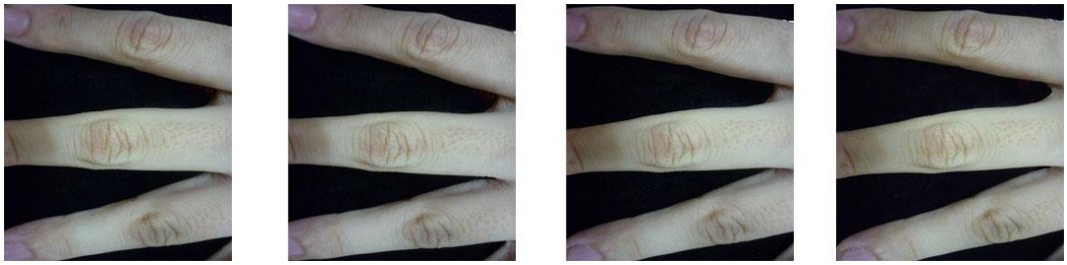
Gambar 4.20 Ruas jari tangan Bayu (sample 1-4)



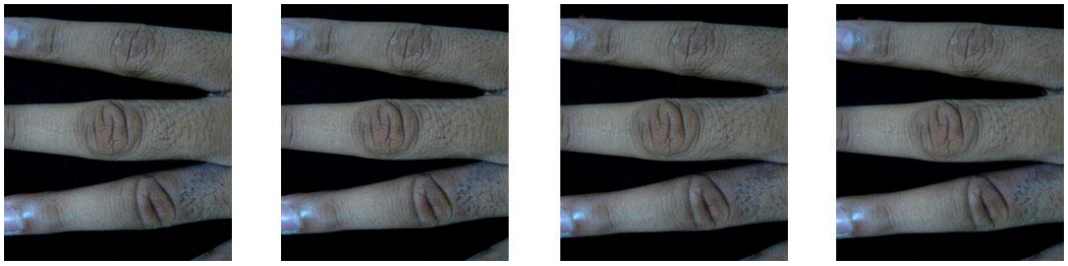
Gambar 4.21 Ruas jari tangan Chandra (sample 1-4)



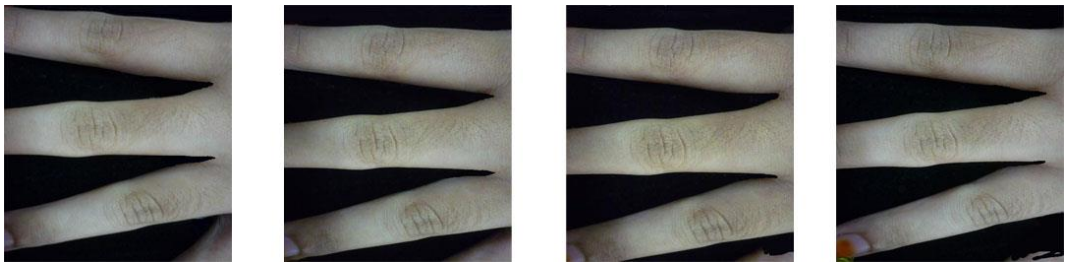
Gambar 4.22 Ruas jari tangan Danika (sample 1-4)



Gambar 4.23 Ruas jari tangan Deby (sample 1-4)



Gambar 4.24 Ruas jari tangan Dimas (sample 1-4)



Gambar 4.25 Ruas jari tangan Dita (sample 1-4)



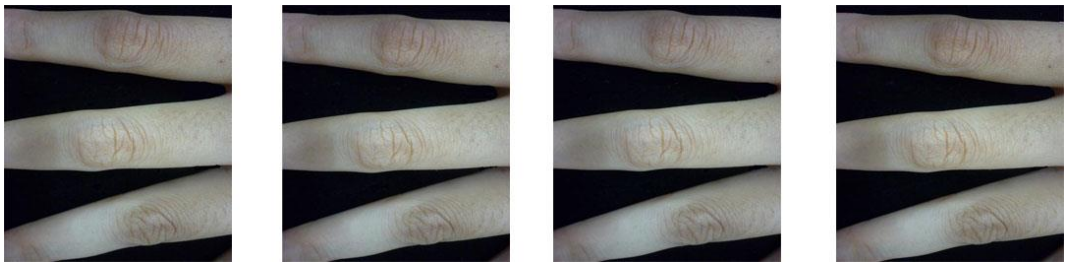
Gambar 4.26 Ruas jari tangan Edu (sample 1-4)



Gambar 4.27 Ruas jari tangan Falaq (sample 1-4)



Gambar 4.28 Ruas jari tangan Firman (sample 1-4)



Gambar 4.29 Ruas jari tangan Helsha (sample 1-4)



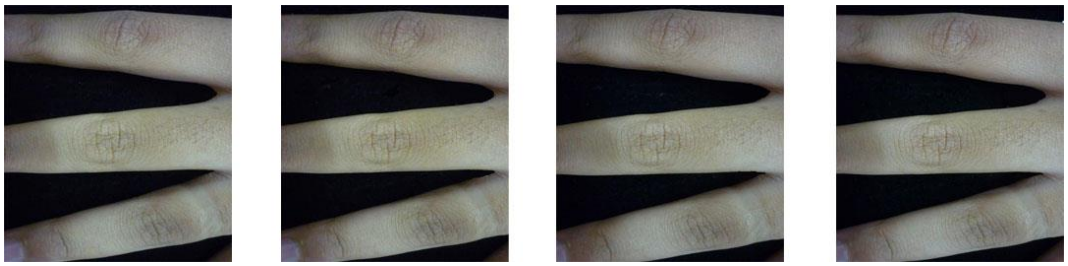
Gambar 4.30 Ruas jari tangan Hendry (sample 1-4)



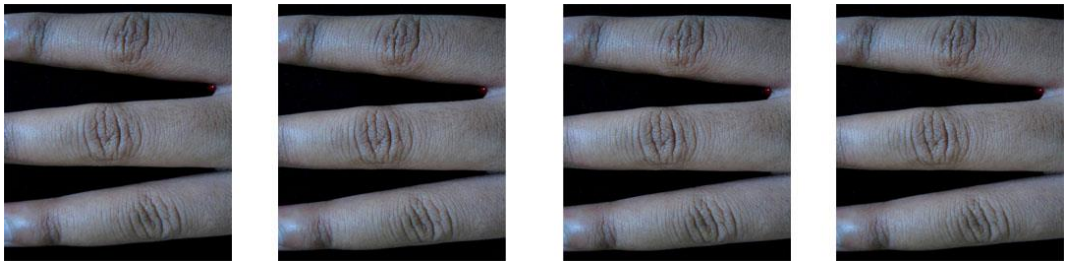
Gambar 4.31 Ruas jari tangan Iva (sample 1-4)



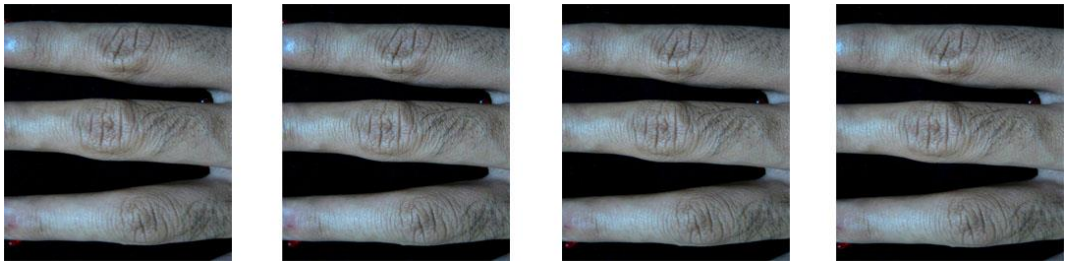
Gambar 4.32 Ruas jari tangan Labib (sample 1-4)



Gambar 4.33 Ruas jari tangan Laras (sample 1-4)



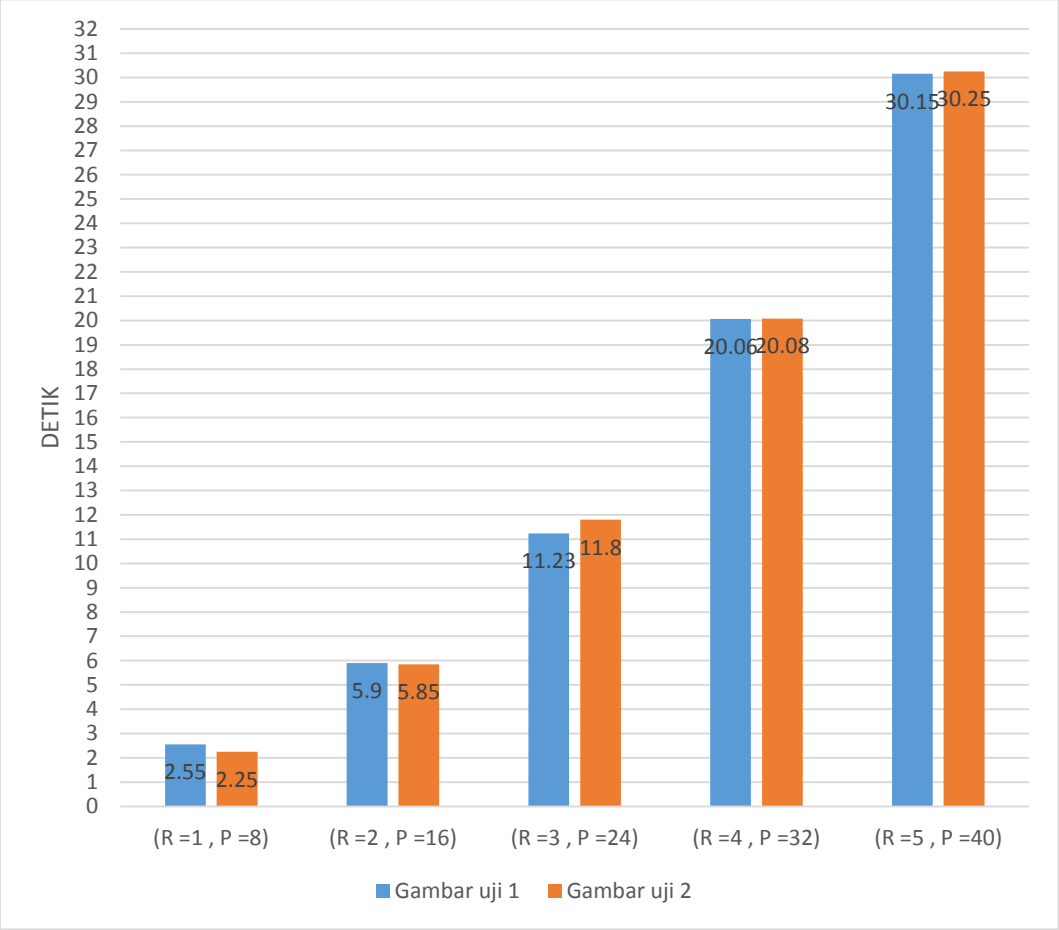
Gambar 4.34 Ruas jari tangan Nunuy (sample 1-4)



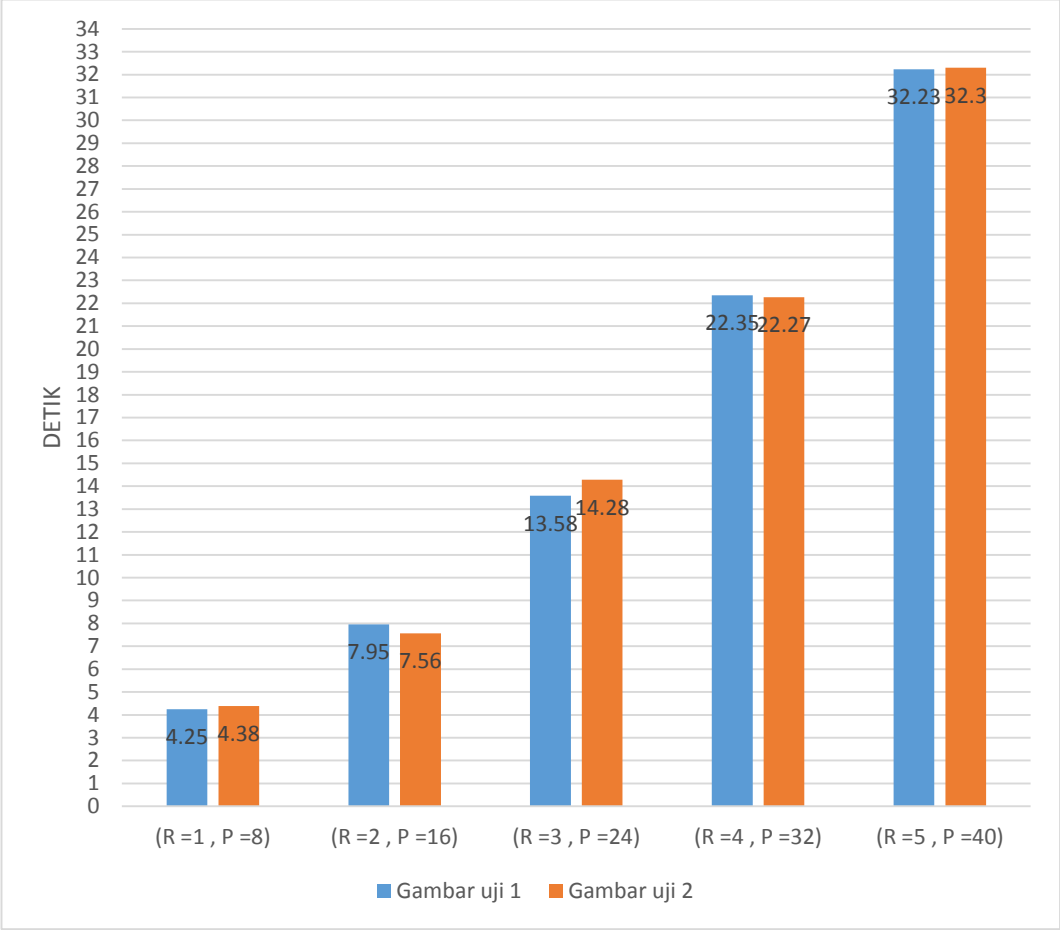
Gambar 4.35 Ruas jari tangan Wara (sample 1-4)

4.2.5.1 Pengujian Performansi

Hasil pengujian performansi berdasarkan kecepatan waktu dalam proses ekstrkasi ciri LBP pattern dan proses keseluruhan komputasi yang tercepat menurut parameter LBP, yaitu *radius* & *sampling point* .Pengujiannya sebagai berikut :



Gambar 4.36 Grafik waktu proses ekstrkasi ciri LBP pattern



Gambar 4.37 Grafik waktu proses keseluruhan komputasi

Pengujian dilakukan terhadap 2 gambar uji yang gambarnya beresolusi 480 x 640 pixel dan 8 bit menggunakan parameter LBP tersebut dengan perhitungan proses waktu perdetik.

4.2.5.2 Kesimpulan Pengujian Performansi

Dari hasil pengujian performansi menurut Gambar 4.36 dan Gambar 4.37 menyimpulkan bahwa dari pengujian terhadap 2 gambar uji didapatkan kesimpulan bahwa parameter LBP dengan *radius* terkecil prosesnya lebih cepat dan efisien dari pada parameter LBP dengan *radius* lebih besar. Hal ini dikarenakan adanya pengaruh besarnya parameter LBP terhadap waktu komputasi. Semakin besar rentang nilai parameter *radius*, semakin banyak piksel yang harus diproses sehingga membutuhkan waktu yang lama. Jadi untuk mendapatkan performansi yang bagus dan efisien menggunakan parameter LBP dengan *radius* 1 dan *sampling point* 8.

Dengan menggunakan parameter tersebut, aplikasi dapat berjalan dengan cepat dan tentunya dengan performansi yang bagus.

4.2.5.3 Pengujian Keakuratan

Hasil pengujian keakuratan pengenalan ruas jari tangan terhadap gambar sebagai berikut :

a. Pengujian berdasarkan gambar sample 1, 2, 3 & 4 ruas jari tangan dan menggunakan parameter LBP terbaik yaitu r = 1 & p = 8.

Tabel 4.19 Hasil pengujian (a) Adil (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Adil 1a.jpg	0	Teridentifikasi
2	Adil 2b.jpg	0,16	Teridentifikasi
3	Adil 3c.jpg	0,18	Teridentifikasi
4	Adil 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.20 Hasil pengujian (a) Adli (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Adli 1a.jpg	0	Teridentifikasi
2	Adli 2b.jpg	0,1	Teridentifikasi
3	Adli 3c.jpg	0	Teridentifikasi
4	Adli 4d.jpg	0,1	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.21 Hasil pengujian (a) Aul (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Aul 1a.jpg	0	Teridentifikasi
2	Aul 2b.jpg	0	Teridentifikasi
3	Aul 3c.jpg	0	Teridentifikasi
4	Aul 4d.jpg	0,1	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.22 Hasil pengujian (a) Authar (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Authar 1a.jpg	0	Teridentifikasi
2	Authar 2b.jpg	0,16	Teridentifikasi
3	Authar 3c.jpg	0	Teridentifikasi
4	Authar 4d.jpg	0,12	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.23 Hasil pengujian (a) Bayu (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Bayu 1a.jpg	0	Teridentifikasi
2	Bayu 2b.jpg	0	Teridentifikasi
3	Bayu 3c.jpg	0	Teridentifikasi

4	Bayu 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.24 Hasil pengujian (a) Chandra (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Chandra 1a.jpg	0	Teridentifikasi
2	Chandra 2b.jpg	0,24	Teridentifikasi
3	Chandra 3c.jpg	0,17	Teridentifikasi
4	Chandra 4d.jpg	0,24	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.25 Hasil pengujian (a) Danika (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Danika 1a.jpg	0	Teridentifikasi
2	Danika 2b.jpg	0	Teridentifikasi
3	Danika 3c.jpg	0	Teridentifikasi
4	Danika 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.26 Hasil pengujian (a) Deby (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Deby 1a.jpg	0	Teridentifikasi
2	Deby 2b.jpg	0	Teridentifikasi
3	Deby 3c.jpg	0	Teridentifikasi
4	Deby 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.27 Hasil pengujian (a) Dimas (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Dimas 1a.jpg	0	Teridentifikasi
2	Dimas 2b.jpg	0	Teridentifikasi
3	Dimas 3c.jpg	0	Teridentifikasi
4	Dimas 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.28 Hasil pengujian (a) Dita (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Dita 1a.jpg	0	Teridentifikasi
2	Dita 2b.jpg	0	Teridentifikasi
3	Dita 3c.jpg	0	Teridentifikasi

4	Dita 4d.jpg	0,19	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.29 Hasil pengujian (a) Edu (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Edu 1a.jpg	0	Teridentifikasi
2	Edu 2b.jpg	0	Teridentifikasi
3	Edu 3c.jpg	0	Teridentifikasi
4	Edu 4d.jpg	0,28	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.30 Hasil pengujian (a) Falaq (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Falaq 1a.jpg	0	Teridentifikasi
2	Falaq 2b.jpg	0,25	Teridentifikasi
3	Falaq 3c.jpg	2	Gagal
4	Falaq 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 3

Persentase keakuratan : 75%

Tabel 4.31 Hasil pengujian (a) Firman (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Firman 1a.jpg	0	Teridentifikasi
2	Firman 2b.jpg	0	Teridentifikasi
3	Firman 3c.jpg	0	Teridentifikasi
4	Firman 4d.jpg	0,13	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.32 Hasil pengujian (a) Helsha (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Helsha 1a.jpg	0	Teridentifikasi
2	Helsha 2b.jpg	0,1	Teridentifikasi
3	Helsha 3c.jpg	0	Teridentifikasi
4	Helsha 4d.jpg	1	Gagal

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 3

Persentase keakuratan : 75%

Tabel 4.33 Hasil pengujian (a) Hendry (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Hendry 1a.jpg	1	Gagal
2	Hendry 2b.jpg	0	Teridentifikasi
3	Hendry 3c.jpg	0	Teridentifikasi

4	Hendry 4d.jpg	0,14	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 3

Persentase keakuratan : 75%

Tabel 4.34 Hasil pengujian (a) Iva (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Iva 1a.jpg	0	Teridentifikasi
2	Iva 2b.jpg	0	Teridentifikasi
3	Iva 3c.jpg	0,23	Teridentifikasi
4	Iva 4d.jpg	0,3	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.35 Hasil pengujian (a) Labib (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Labib 1a.jpg	0	Teridentifikasi
2	Labib 2b.jpg	0	Teridentifikasi
3	Labib 3c.jpg	0	Teridentifikasi
4	Labib 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.36 Hasil pengujian (a) Laras (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Laras 1a.jpg	3	Gagal
2	Laras 2b.jpg	0,2	Teridentifikasi
3	Laras 3c.jpg	0	Teridentifikasi
4	Laras 4d.jpg	0,27	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 3

Persentase keakuratan : 75%

Tabel 4.37 Hasil pengujian (a) Nunuy (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Nunuy 1a.jpg	0	Teridentifikasi
2	Nunuy 2b.jpg	0,1	Teridentifikasi
3	Nunuy 3c.jpg	0,13	Teridentifikasi
4	Nunuy 4d.jpg	0	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

Tabel 4.38 Hasil pengujian (a) Wara (sample 1-4)

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	Wara 1a.jpg	0	Teridentifikasi
2	Wara 2b.jpg	0,13	Teridentifikasi
3	Wara 3c.jpg	0	Teridentifikasi

4	Wara 4d.jpg	0,17	Teridentifikasi

Banyak pengujian : 4

Jumlah pengenalan gambar yang dikenali : 4

Persentase keakuratan : 100%

b. Pengujian berdasarkan parameter LBP yang berbeda-beda untuk diketahui hasil gambar teridentifikasi atau tidaknya.

Tabel 4.39 Hasil pengujian dengan parameter berbeda-beda

No.	Gambar Uji	Nilai Euclidean Distance	Hasil
1	- Danika 2b.jpg - (R=2, P=16)	0	Teridentifikasi
2	- Iva 3c.jpg - (R=2, P=16)	0,23	Teridentifikasi
3	- Firman 4d.jpg - (R=3, P=24)	0,13	Teridentifikasi
4	- Labib 1a.jpg - (R=3, P=24)	0	Teridentifikasi
5	- Deby 2b.jpg - (R=4, P=32)	0	Teridentifikasi
6	- Helsha 3c.jpg - (R=4, P=32)	0	Teridentifikasi
7	- Nunuy 2b.jpg - (R=5, P=40)	0,1	Teridentifikasi
8	- Wara 3c.jpg - (R=5, P=40)	0	Teridentifikasi

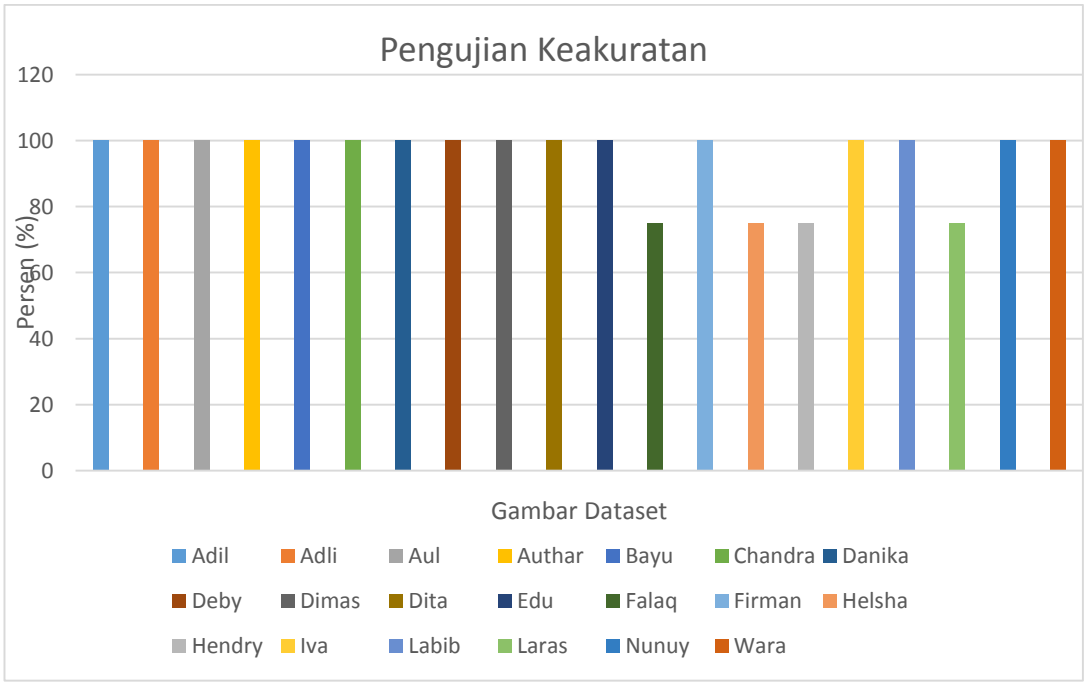
c. Pengujian berdasarkan data masukan salah atau tidak sesuai / belum adanya data latihnya.

Tabel 4.40 Hasil pengujian dari data masukan yang tidak sesuai

No.	Gambar Uji	Hasil	Keterangan
1	- OSM.png - 1360 x 768 px	Gagal	- Dikarenakan gambar bukan sesuai ukuran piksel dan berbeda objek.
2	- lilin.png - 1354 x 580 px	Gagal	- Dikarenakan gambar bukan sesuai ukuran piksel dan berbeda objek.
3	- chandra.jpg - 280 x 340 px	Gagal	- Dikarenakan gambar bukan sesuai ukuran piksel walaupun objeknya sesuai.
4	- struktur.jpg - 960 x 720 px	Gagal	- Dikarenakan gambar bukan sesuai ukuran piksel dan berbeda objek.

4.2.5.4 Kesimpulan Pengujian Keakuratan

Berdasarkan hasil pengujian keakuratan yang telah dilakukan dan terdapat hasil data sebagai berikut :



Gambar 4.38 Grafik data seluruh hasil pengujian keakuratan

Kesimpulan dari tabel hasil nilai euclidean 0 sama dengan teridentifikasi, sedangkan nilai 1 keatas memungkinkan hasilnya tidak teridentifikasi ataupun gagal.

Jumlah Rata-Rata	= Hasil perjumlahan seluruh persentase pengujian / banyaknya pengujian.
Jumlah Rata-Rata	= 1900% / 20
	= **95%**

Dari data tersebut dapat diambil kesimpulan bahwa persentase rata-ratanya untuk semua gambar pada pengujian yaitu : **95 %**