

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan pengujian terhadap sistem pengenalan yang telah dibuat. Dengan memanfaatkan PC (*personal computer*) yang telah terinstal aplikasi pengenalan wajah dan *mobile robot omnidirectional* yang digunakan untuk melakukan pencarian orang.

4.1. Pengujian Koneksi Robotino

Pengujian koneksi Robotino dilakukan dengan menguji jaringan *wireless* pada PC yang dihubungkan dengan *access point* Robotino. Hal tersebut untuk diketahui apakah dapat terhubung, dan saat program dijalankan apakah *console* sudah mampu mengakses Robotino.

4.1.1. Tujuan

Tujuan pengujian ini yaitu mengetahui apakah Robotino sudah mampu terhubung dengan PC. Mengetahui apakah *console* yang digunakan sudah mampu mengakses Robotino, sehingga dapat disimpulkan apakah Robotino telah terintegrasi dengan PC.

4.1.2. Alat yang Digunakan

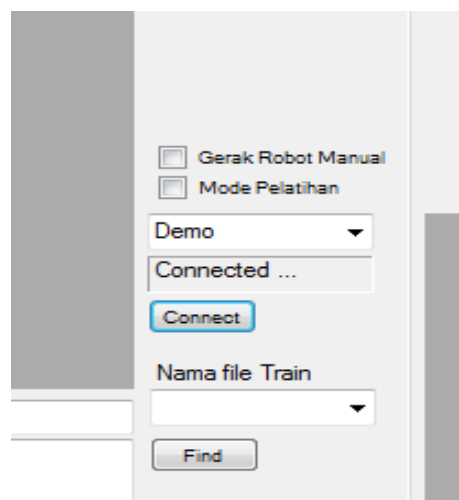
1. Robotino
2. Microsoft Visual C# 2012
3. *Personal Computer* (PC)

4.1.3. Prosedur Pengujian

1. Menyalakan Robotino
2. Menghubungkan koneksi *wireless* dari PC ke *access point* Robotino.
3. Menjalankan program *console* pada Microsoft Visual C# 2012
4. Menghubungkan *console* dengan Robotino sesuai IP yang ditentukan (128.0.0.1 untuk simulasi dan 172.16.1.1 untuk Robotino)
5. Menjalankan program *manual control* untuk memastikan bahwa koneksi sudah terhubung dengan baik.

4.1.4. Hasil Pengujian

Robotino sudah dapat terhubung ke PC dengan media *wireless* dengan baik. Terlihat pada Gambar 4.1, saat program *console* dijalankan terdapat 2 pilihan koneksi yaitu langsung ke Robotino atau simulasi. Pada simulasi seluruh fungsi Robotino dapat digunakan seperti Robotino yang sesungguhnya, sehingga jika dihubungkan koneksi pada simulasi dan berhasil, maka sudah dapat dipastikan koneksi yang dihubungkan ke Robotino secara langsung juga akan berhasil.



Gambar 4.1 *Main form* saat melakukan koneksi dengan Robotino

Untuk pengujian ke Robotino secara langsung dapat dilihat pada LCD yang ditampilkan pada Robotino bahwa koneksi antara Robotino dengan PC sudah terhubung. Untuk lebih jelasnya terlihat pada Gambar 4.2.

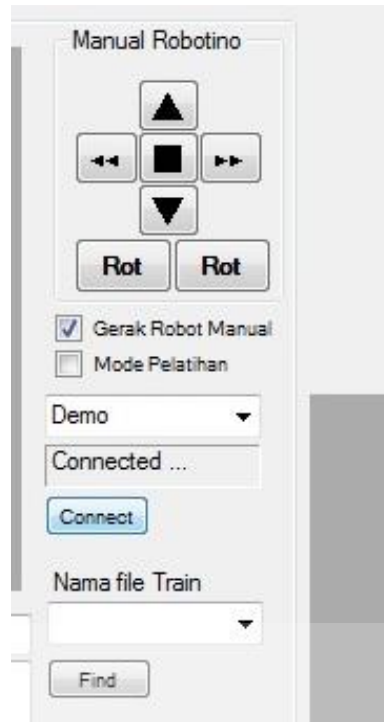


Gambar 4.2 *Console* Terhubung Dengan Robotino

Dari hasil pengujian diatas Robotino sudah mampu terhubung dengan komputer PC, dan *console* yang digunakan sudah mampu mengakses Robotino, sehingga dapat disimpulkan Robotino telah terintegrasi dengan PC.

4.2. Pengujian Pergerakan Robotino

Pengujian pergerakan Robotino dilakukan dengan memanfaatkan *omni-directional drive* pada Robotino. Yaitu dengan menguji fungsi *setVelocity* saat diisi dengan parameter tertentu pada *console*, apakah Robotino akan bergerak sesuai dengan yang diharapkan.



Gambar 4.3 Mode gerak manual untuk Robotino

Pada Gambar 4.3 dapat dilihat terdapat beberapa *button* untuk memerintahkan Robotino. Pada mode tersebut ada perintah maju, mundur, berhenti, ke kanan, ke kiri, berputar ke kanan, dan berputar ke kiri. Jika koneksi antara Robotino dengan komputer sudah dilakukan untuk melakukan pengecekan bisa menggunakan mode ini.

4.2.1. Tujuan

Tujuan pengujian pergerakan Robotino memanfaatkan *omni-directional drive* Robotino adalah untuk mengetahui arah pergerakan Robotino dengan mengatur parameter untuk pergerakan Robotino apakah sesuai dengan yang diinginkan. Maka hasil yang didapatkan akan menunjukkan arah pergerakan Robotino.

4.2.2. Alat yang Digunakan

1. Robotino
2. Microsoft Visual Csharp 2012
3. *Personal Computer (PC)*

4.2.3. Prosedur Pengujian

1. Menyalakan Robotino
2. Menghubungkan koneksi *wireless* dari PC ke *access point* Robotino
3. Menjalankan program *console* pada Microsoft Visual CSharp 2012
4. Menghubungkan PC dengan Robotino sesuai dengan pemilihan dari *user* pada program (128.0.0.1 untuk simulasi dan 172.16.1.1 untuk Robotino)
5. Menjalankan program dan melakukan pemilihan mode *manual* Robotino
6. Melihat arah pergerakan Robotino, sesuai dengan *input user*








4.2.4. Hasil Pengujian

Hasil dari pengujian ini adalah mengetahui arah pergerakan Robotino melalui parameter-parameter yang ada yaitu x,y dan omega. Untuk pergerakan Robotino dengan arah tertentu dilakukan konfigurasi parameter sesuai Tabel 4.1.

Untuk melakukan pergerakan menggunakan perintah,

```
setVelocity(Vx, Vy , Omega);
```

Tabel 4.1 Pergerakan Robotino

Button	Parameter SetVelocity			Hasil Pergerakan Robotino
	Vx	Vy	Omega	
	200	0	0	Maju
	-200	0	0	Mundur
	0	-200	0	Kanan
	0	200	0	Kiri
	0	0	0	Berhenti
	0	0	20	Berputar ke Kiri
	0	0	-20	Berputar ke Kanan

Dengan demikian dapat disimpulkan bahwa *omni-directional drive* pada Robotino dengan parameter setVelocity yang diisi sesuai tabel dan pengujian di atas sudah dapat berjalan dengan baik dan sesuai dengan arah yang diharapkan.

4.3. Pengujian *Streaming* Citra Melalui Kamera Robotino

Pengujian *streaming* gambar Robotino dilakukan dengan melihat data gambar yang tampil pada *window* apakah sesuai dengan yang terlihat pada kamera Robotino. Dan setelah itu melihat apakah data gambar yang dikeluarkan mampu diolah dan ditampilkan secara *streaming*.

4.3.1. Tujuan

Tujuan pengujian ini yaitu untuk mengetahui apakah aplikasi sudah mampu menampilkan data gambar dari *webcam* Robotino ke *window* pada PC, dan untuk gambar yang ditampilkan apakah sudah *streaming*.

4.3.2. Alat yang Digunakan

1. Robotino
2. *Webcam*
3. Microsoft Visual CSharp 2012
4. Personal Computer (PC)

4.3.3. Prosedur Pengujian

1. Menyalakan Robotino
2. Menghubungkan koneksi *wireless* dari PC ke *access point* Robotino
3. Menjalankan program pada Microsoft Visual CSharp 2012
4. Menghubungkan *console* dengan Robotino sesuai IP yang ditentukan (128.0.0.1 untuk simulasi dan 172.16.1.1 untuk Robotino)
5. Menjalankan program untuk mengakses data gambar pada *webcam* Robotino
6. Melihat data gambar hasil *output* dari Robotino pada program

4.3.4. Hasil Pengujian

Setelah melakukan pengujian sesuai dengan prosedur di atas berikut adalah gambar yang didapatkan dari *webcam* Robotino pada Gambar 4.4.



Gambar 4.4 Gambar *streaming* kamera Robotino

Dengan melihat hasil pengujian di atas, dapat disimpulkan bahwa aplikasi sudah mampu mendapatkan data citra pada *webcam* Robotino dan mampu menampilkannya secara *streaming* kepada PC.

4.4. Pengujian *Streaming* Citra Melalui Kamera PC

Pengujian *streaming* ini dilakukan dengan mengintegrasikan Microsoft Visual CSharp melalui library EmguCV. Yaitu untuk memanggil serta menjalankan *console* kamera PC tersebut. Dengan menggunakan fungsi `Capture _cap_web` untuk melakukan inisialisasi kamera yang terintegrasi pada PC. Lalu menampilkan hasil *output* dari kamera dengan fungsi `_cap_web.QueryFrame()`.

4.4.1. Tujuan

Tujuan pengujian ini yaitu untuk mengetahui apakah aplikasi sudah mampu menampilkan data citra dari kamera PC ke aplikasi pada Visual CSharp dan apakah dapat langsung diproses oleh program.

4.4.2. Alat yang Digunakan

1. Microsoft Visual CSharp 2012
2. Personal Computer (PC) dengan *webcam*

4.4.3. Prosedur Pengujian

1. Menjalankan program *console* pada Microsoft Visual CSharp 2012
2. Menjalankan program untuk mengakses data citra pada kamera PC tersebut
3. Melihat hasil data citra pada *window*

4.4.4. Hasil Pengujian

Setelah melakukan pengujian sesuai dengan prosedur diatas berikut adalah gambar yang didapatkan dari kamera PC pada Gambar 4.5.



Gambar 4.5 Gambar *streaming* dengan kamera PC

4.5. Pengujian Deteksi Wajah

Pengujian deteksi wajah dilakukan pada hasil kamera yang ditampilkan pada program. Pendeteksian ini dilakukan dengan menggunakan *library* yang telah disediakan, yaitu dengan menggunakan fungsi `HaarObjectDetector Object_ut`. Pengujian ini akan dilakukan berdasarkan pose wajah, jarak, dan ekspresi wajah yang tertangkap oleh kamera.

4.5.1. Tujuan

Pengujian ini bertujuan untuk menguji apakah program sudah dapat mendeteksi wajah yang tertangkap dari hasil kamera. Dengan berbagai model pengujian yang dilakukan ini dimaksudkan untuk meminimalisir *error* yang terjadi pada saat *running* program, mulai dari saat *training* maupun *recognition*.

4.5.2. Alat yang Digunakan

1. *Webcam*
2. Microsoft Visual CSharp 2012
3. *Personal Computer* (PC)
4. Objek wajah manusia

4.5.3. Prosedur Pengujian

1. Menjalankan program pada Microsoft Visual CSharp 2012
2. Menjalankan program untuk mengakses *webcam*
3. Menjalankan program untuk proses deteksi wajah

4.5.4. Hasil Pengujian

Pengujian dengan berbagai pose dapat dilihat hasilnya pada Gambar 4.5.



Gambar 4.6 Pendeteksian wajah berdasarkan pose

Pada Gambar 4.6 dapat dilihat ada beberapa pose wajah yang tidak terdeteksi, hal ini dikarenakan oleh pose dari wajah, seperti pada Tabel 4.2.

Tabel 4.2 Hasil pendeteksian wajah dengan beberapa pose

Percobaan	Pose Wajah	Hasil Deteksi
1	Miring Kiri	Tidak Terdeteksi
2	Miring Kanan	Tidak Terdeteksi
3	Menengok Kekanan	Tidak Terdeteksi
4	Menengok Kekiri	Tidak Terdeteksi
5	Mendongak Ke atas	Tidak Terdeteksi
6	Mendongak Ke bawah	Tidak Terdeteksi
7	Melihat Ke depan	Terdeteksi
8	Serong Kanan	Terdeteksi
9	Serong Kiri	Terdeteksi
10	Melihat Kedepan agak kebawah	Terdeteksi

Pengujian pada berbagai jarak dalam satuan centimeter dapat dilihat pada Gambar 4.7.



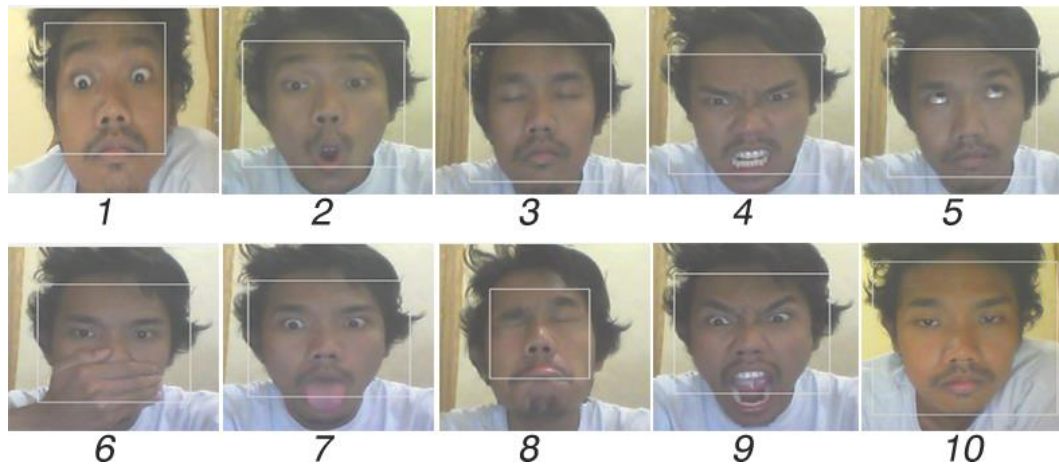
Gambar 4.7 Pendeteksian wajah pada berbagai jarak

Pada Gambar 4.7 dapat dilihat pendeteksian wajah yang dilakukan pada berbagai jarak, hasil pendeteksian seperti pada Tabel 4.3.

Tabel 4.3 Hasil pendeteksian wajah dalam berbagai jarak

Percobaan	Jarak kamera dengan objek	Hasil Deteksi	
		Terdeteksi	Performa
1	40 cm	Ya	Baik
2	80 cm	Ya	Baik
3	120 cm	Ya	Baik
4	160 cm	Ya	Baik
5	200 cm	Ya	Kurang baik
6	240 cm	Ya	Buruk
7	280 cm	Tidak	Buruk
8	320 cm	Tidak	Buruk

Pengujian selanjutnya dengan melakukan pendeteksian wajah dengan berbagai ekspresi wajah



Gambar 4.8 Pendeteksian wajah dengan berbagai ekspresi wajah

Pada Gambar 4.8 dapat dilihat bahwa semua ekspresi wajah yang diujikan dapat dideteksi selama wajah masih dalam jarak yang bisa dideteksi dan pose wajah juga dalam batas pose yang bisa terdeteksi.

4.6. Pengujian Pengolahan Citra

Pengujian berikut adalah pengujian data citra pada hasil *capture* dari pendeteksian wajah. Pengolahan citra yang dilakukan pada proses adalah sebatas melakukan *resize* gambar wajah, konversi warna dari RGB ke *grayscale* dan proses perataan gambar dengan histogram ekualisasi. Gambar hasil dari proses *capture* akan diseragamkan ukurannya, di ubah menjadi gambar wajah dengan ukuran 100x100. Lalu gambar akan di konversikan ke dalam ruang warna *grayscale* dan proses selanjutnya melakukan perataan gambar dengan proses histogram ekualisasi.

4.6.1. Tujuan

Pengujian ini bertujuan untuk mempersiapkan gambar yang dihasilkan dari proses *capture* wajah agar menjadi sebuah gambar yang siap diproses pada proses

selanjutnya (PCA). Agar memudahkan dalam melakukan perhitungan nilai ciri dari tiap wajah yang dilatihkan.

4.6.2. Alat yang Digunakan

1. *Webcam*
2. Microsoft Visual CSharp 2012
3. *Personal Computer (PC)*
4. Objek wajah manusia

4.6.3. Prosedur Pengujian

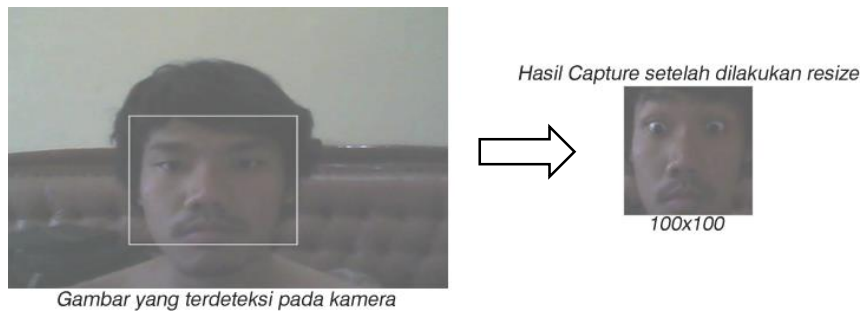
4. Menjalankan program pada Microsoft Visual CSharp 2012
5. Menjalankan program untuk mengakses *webcam*
6. Menjalankan program untuk proses *capture*
7. Melakukan proses pengolahan citra pada program

4.6.4. Hasil Pengujian

Untuk proses pengolahan citra, pertama-tama gambar hasil *capture* akan *resize* dengan menggunakan perintah berikut:

```
Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
```

Citra hasil *resize* dari baris perintah di atas dapat dilihat seperti pada Gambar 4.9.



Gambar 4.9 Hasil resize gambar dengan resolusi pixel 100x100

Setelah melakukan proses *resize* selanjutnya gambar akan diubah ke dalam ruang warna *grayscale*, dengan menggunakan perintah

```
Convert<Gray, byte>();
```

Hasil dari proses *grayscale* dapat dilihat pada Gambar 4.10.

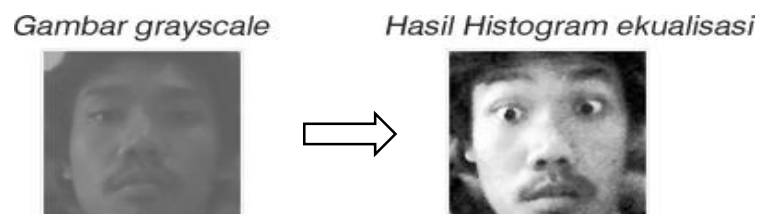


Gambar 4.10 Hasil konversi RGB ke *grayscale*

Proses pengolahan citra selanjutnya adalah melakukan perataan gambar dengan histogram ekualisasi, dengan perintah pada program

```
gray_img._EqualizeHist();
```

Hasilnya akan seperti gambar 4.11.



Gambar 4.11 Hasil perataan gambar dengan histogram ekualisasi

Setelah melalui proses perataan gambar seperti pada Gambar 4.11, akan didapat gambar yang lebih jelas daripada gambar sebelumnya, hal ini bertujuan

untuk memudahkan dalam melakukan perhitungan pada proses selanjutnya. Gambar ini yang akan menjadi *database* untuk tiap orang yang melakukan pelatihan.

4.7. Pengujian Metode PCA pada Proses *Training*

Pengujian pada penerapan metode *principal component analysis* dalam melakukan proses pembuatan *database* pengenalan wajah. Apakah sudah didapatkan nilai-nilai yang dibutuhkan dari proses ini, antara lain rata-rata, nilai eigen, nilai eigen vektor, nilai eigenface dan nilai eigenspace.

4.7.1. Tujuan

Tujuan Pengujian ini adalah untuk melihat apakah aplikasi telah dapat mengambil nilai-nilai penting yang dibutuhkan pada metode *principal component analysis*.

4.7.2. Alat yang Digunakan

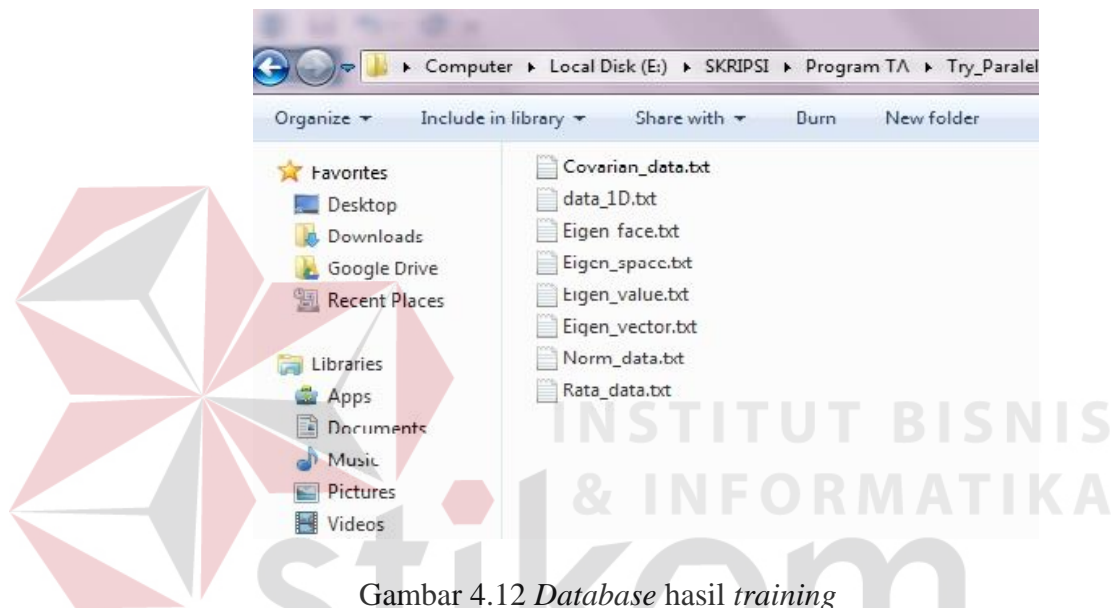
1. Microsoft Visual CSharp 2012
2. *Personal Computer* (PC) dengan kamera web
3. Objek wajah manusia

4.7.3. Prosedur Pengujian

1. Menjalankan program *console* pada Microsoft Visual CSharp 2012
2. Menjalankan program untuk mengakses data citra pada kamera PC tersebut
3. Melakukan proses *training* dan menyimpan hasil datanya
4. Melakukan pengecekan *database* pada direktori yang telah ditentukan

4.7.4. Hasil Pengujian

Hasil dari pengujian ini untuk melakukan pengecekan data dari nilai yang telah didapatkan dari proses *training* menggunakan metode *principal component analysis*. Dalam pengujian ini data yang disimpan berupa file berekstensi .txt, yang berisi data dari gambar yang dilatihkan seperti pada Gambar 4.12.



Gambar 4.12 Database hasil training

Database dari tiap orang yang telah melakukan proses *training* akan berisi data_1D.txt, Rata_data.txt, Norm_data.txt, Covarian_data.txt, Eigen_value.txt, Eigen_vector.txt, Eigen_face.txt, dan Eigen_space.txt.

4.8. Pengujian Metode PCA pada Proses *Recognition*

Pengujian metode *principal component analysis* dalam melakukan proses pengenalan pada wajah meliputi pengujian terhadap ekspresi wajah, jarak wajah terhadap kamera, dan intensitas cahaya. Pengujian ini dilakukan pada objek wajah orang yang telah melakukan *training* sebelumnya.

4.8.1. Tujuan

Tujuan Pengujian ini adalah untuk melihat tingkat keakurasian dari metode *principal component analysis* dalam melakukan proses pengenalan wajah. Dari hasil ini juga dapat disimpulkan kelemahan dan kelebihan dari metode ini.

4.8.2. Alat yang Digunakan

1. Microsoft Visual CSharp 2012
2. *Personal Computer* (PC) dengan kamera web
3. Objek wajah manusia
4. *Lux* meter (sensor intensitas cahaya)

4.8.3. Prosedur Pengujian











1. Menjalankan program *console* pada Microsoft Visual CSharp 2012
2. Menjalankan program untuk mengakses data citra pada kamera PC tersebut
3. Melakukan proses *training* dan menyimpan hasil datanya
4. Mengakses Robotino SIM Demo
5. Mengatur sensor intensitas cahaya (*lux* meter/*light* meter)
6. Melakukan proses *recognition* pada wajah yang telah melalui proses *training*









4.8.4. Hasil Pengujian

Hasil dari pengujian ini adalah mengetahui tingkat keakurasian dari metode *principal component analysis*. Dalam pengujian ini pengenalan wajah akan dilakukan dengan beberapa kondisi yang berbeda mulai ekspresi wajah, jarak, dan

intensitas cahaya. Pada Tabel 4.2 adalah tabel hasil dari beberapa wajah yang telah melakukan proses *training* sebelumnya.

Tabel 4.2 Daftar tabel pengenalan dengan berbagai ekspresi pada *database* orang yang telah melakukan *training*

Wajah di Database	Objek tes	Keterangan			
		Bowo	Galang	Jee	Keterangan
		84,65920 92357824	39,43995 60482141	84,62087 86119562	Status : Benar Jarak terpendek : 39,439956048 2141
		83,03765 90604648	52,93491 08317882	84,38736 38563229	Status : Benar Jarak terpendek : 52,934910831 7882
		89,35869 30836508	47,76639 29824596	82,04111 54335979	Status : Benar Jarak terpendek : 47,766392982 4596
		88,10460 82986291	60,82494 34373827	85,15521 80793184	Status : Benar Jarak terpendek : 60,824943437 3827
		85,95938 0579912	55,21491 51099181	82,34061 60865697	Status : Benar Jarak terpendek : 55,214915109 9181
		61,59088 10713461	89,24607 33237839	76,47684 67306125	Status : Benar Jarak terpendek : 61,590881071 3461
		43,73644 81911998	88,11359 44234693	87,41430 48006648	Status : Benar Jarak terpendek : 43,736448191 1998
		38,06888 07733053	87,83284 84224855	85,70523 89931775	Status : Benar Jarak terpendek :

					38,068880773 3053
		43,87608 79228283	88,03267 7960998	87,50052 13848261	Status : Benar Jarak terpendek : 43,876087922 8283
		64,57066 9816052	98,32349 3627572	96,80488 43585893	Status : Benar Jarak terpendek : 64,570669816 052
		91,67433 70500997	82,53533 09724962	51,35263 96831101	Status : Benar Jarak terpendek : 51,352639683 1101
		92,67433 74347997	89,24607 33237839	41,24586 91237101	Status : Benar Jarak terpendek : 41,245869123 7101
		90,67389 14719433	87,26900 35271938	48,72557 0167616	Status : Benar Jarak terpendek : 48,725570167 616
		89,80552 27600818	89,95884 89233631	40,98783 60035947	Status : Benar Jarak terpendek : 40,987836003 5947
		91,31229 57112395	85,53537 71436032	59,40922 41547155	Status : Benar Jarak terpendek : 59,409224154 7155

Pada Tabel 4.2 adalah beberapa orang yang telah melakukan proses *training*. Tiap orang akan memiliki 10 gambar dari wajah mereka, dengan resolusi 100 x 100 tiap gambar. Dari satu set gambar orang (10 gambar wajah) seperti pada Tabel 4.2 akan dihasilkan sebuah *database* ciri dari orang tersebut. Pada Tabel 4.2

dapat dilihat hasil pengenalan wajah dengan berbagai ekspresi wajah dari masing-masing orang yang telah melakukan *training*.

Tabel 4.3 Hasil pengenalan wajah dalam jarak tertentu

Jarak	Banyak Percobaan	Objek Uji	Berhasil Sebanyak	Prosentase Keberhasilan
40 cm	30	Orang 1	29	96,66 %
		Orang 2	27	90 %
		Orang 3	28	93,33%
80 cm	30	Orang 1	27	90%
		Orang 2	30	100%
		Orang 3	27	90%
120 cm	30	Orang 1	23	76,66%
		Orang 2	26	86,66%
		Orang 3	25	83,33%
160cm	30	Orang 1	24	80%
		Orang 2	25	83,33%
		Orang 3	25	83,33%
200 cm	30	Orang 1	5	16,66%
		Orang 2	7	23,33%
		Orang 3	15	50%

Pada Tabel 4.5 adalah pengujian proses *recognition* dalam berbagai kondisi pencahayaan yang diukur melalui intensitas cahayanya. Untuk mengukur intensitas cahaya digunakan alat pengukur intensitas cahaya (*lux meter/light meter*). Pada *lux meter* akan diatur untuk menggunakan mode Ev. Dimana hasil pembacaan sensor akan dikonversi menjadi satuan intensitas cahaya (Lux) sesuai dengan Tabel 4.4.

Tabel 4.4 Tabel konversi nilai Ev ke Lux (Sekonic)

EV	Lux	EV	Lux	EV	Lux	EV	Lux
0.0	2.5	5.0	80	10.0	2600	15.0	82000
0.5	3.5	5.5	110	10.5	3600	15.5	120000
1.0	5.0	6.0	160	11.0	5100	16.0	160000
1.5	7.1	6.5	230	11.5	7200	16.5	230000
2.0	10	7.0	320	12.0	10000	17.0	330000
2.5	14	7.5	450	12.5	14000	17.5	460000
3.0	20	8.0	640	13.0	20000	18.0	660000
3.5	28	8.5	910	13.5	29000	18.5	930000
4.0	40	9.0	1300	14.0	41000	19.0	1300000
4.5	57	9.5	1800	14.5	58000	19.5	1900000

Tabel 4.5 Hasil pengenalan wajah pada beberapa kondisi pencahayaan saat melakukan *training* dan saat melakukan *recognition*

Kondisi pencahayaan saat <i>training</i>		Kondisi pencahayaan saat <i>recognition</i>		Hasil dari 20 kali uji	Selisih Intensitas Cahaya <i>Training - Recognition</i>
Ev	Lux	Ev	Lux		
5 Ev	80 Lux	6 Ev	160 Lux	Benar : 20 Salah : 0	80 Ev
		9 Ev	1300 Lux	Benar : 10 Salah : 10	1220 Ev
		12 Ev	10000 Lux	Benar : 0 Salah : 20	9920 Ev
8 Ev	640 Lux	6 Ev	160 Lux	Benar : 10 Salah : 10	480 Ev
		9 Ev	1300 Lux	Benar : 20 Salah : 0	660 Ev
		12 Ev	10000 Lux	Benar : 0 Salah : 20	9360 Ev
12 Ev	10000 Lux	6 Ev	160 Lux	Benar : 0 Salah : 20	9840 Ev

		9 Ev	1300 Lux	Benar : 5 Salah : 15	8700 Ev
		12 Ev	10000 Lux	Benar : 18 Salah : 2	0 Ev

Pada Tabel 4.4 dapat dilihat perbedaan intensitas cahaya yang diukur menggunakan *lux* meter pada proses *training* dan *recognition* akan mempengaruhi pada hasil *recognition* dari tiap wajah.

Dengan melihat hasil pada Tabel 4.2 sampai dengan Tabel 4.4, dapat disimpulkan apakah metode *principal component analysis* sudah dapat melakukan proses pengenalan dengan baik, dengan prosentase keberhasilan 100 % pada pengujian terhadap ekspresi wajah, 76,22 % untuk pengujian terhadap jarak dan prosentase keberhasilan sekitar 90 – 100 % pada pengujian terhadap intensitas cahaya selama selisih perbedaan antara gambar *training* dan *recognition* antara 0-1300 lux.

4.9. Evaluasi Sistem secara Keseluruhan

Pengujian terakhir adalah pengujian sistem secara keseluruhan dari awal hingga akhir, dimana pengujian dilakukan dengan menjalankan aplikasi secara keseluruhan. Mulai dari proses *training*, sampai proses *recognition*. Pertama-tama sistem ini akan melakukan *training* wajah pada beberapa orang. Setelah itu sistem ini akan menerima perintah dari *user* berupa sebuah inputan nama orang yang sesuai dengan nama yang ada pada *database*, lalu Robotino akan mulai berjalan untuk melakukan pencarian wajah pada tiap orang wajahnya terdeteksi. Robotino akan berjalan sesuai dengan posisi wajah yang terdeteksi, disini robot akan memposisikan wajah tepat ditengah *picturebox*, jika wajah berada dipojok sebelah

kiri maka Robotino akan bergerak ke kanan agar wajah yang terdeteksi berada ditengah dan begitu sebaliknya. Apabila wajah belum tepat berada ditengah dan jarak belum sesuai dengan jarak yang telah diatur maka proses *recognition* tidak akan dilakukan sampai wajah sudah berada diposisi yang telah ditentukan. Setelah wajah berada diposisi yang telah ditentukan, tiap wajah yang terdeteksi akan dicocokkan datanya dengan data yang sedang dicari, apakah sama atau tidak. Robotino akan berhenti setelah menemukan orang yang namanya sesuai dengan inputan *user* tadi.

4.9.1. Tujuan

Tujuan evaluasi sistem ini adalah untuk mengetahui sistem pada aplikasi apakah sudah dapat berjalan sesuai dengan yang diharapkan. Apakah sistem dapat menemukan wajah dari orang yang dicari, sesuai dengan inputan dari *user*.

4.9.2. Alat yang Digunakan

1. Robotino
2. Microsoft Visual CSharp 2012
3. *Personal Computer* (PC) dengan kamera web
4. Objek wajah dari beberapa orang

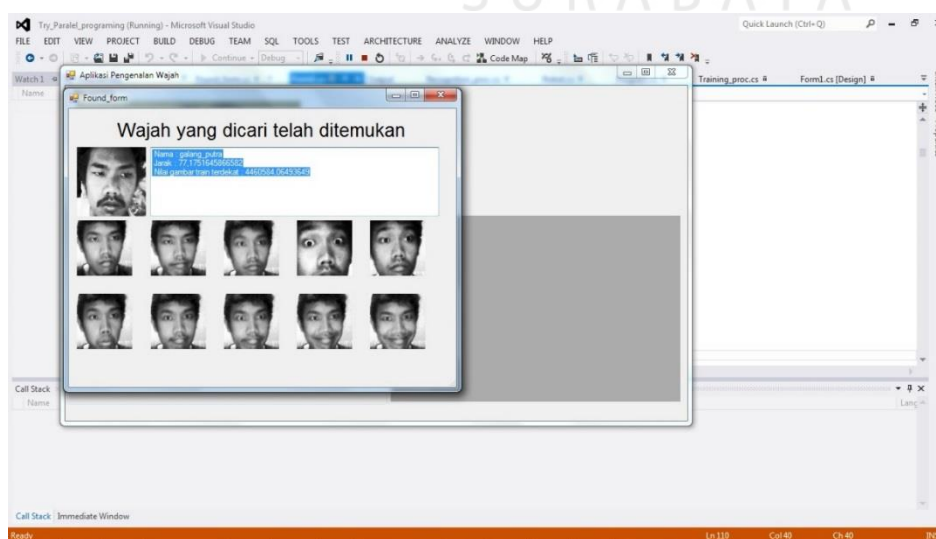
4.9.3. Prosedur Pengujian

1. Menyalakan Robotino
2. Menghubungkan koneksi *wireless* dari PC ke *access point* Robotino
3. Menjalankan program pada Microsoft Visual CSharp 2012

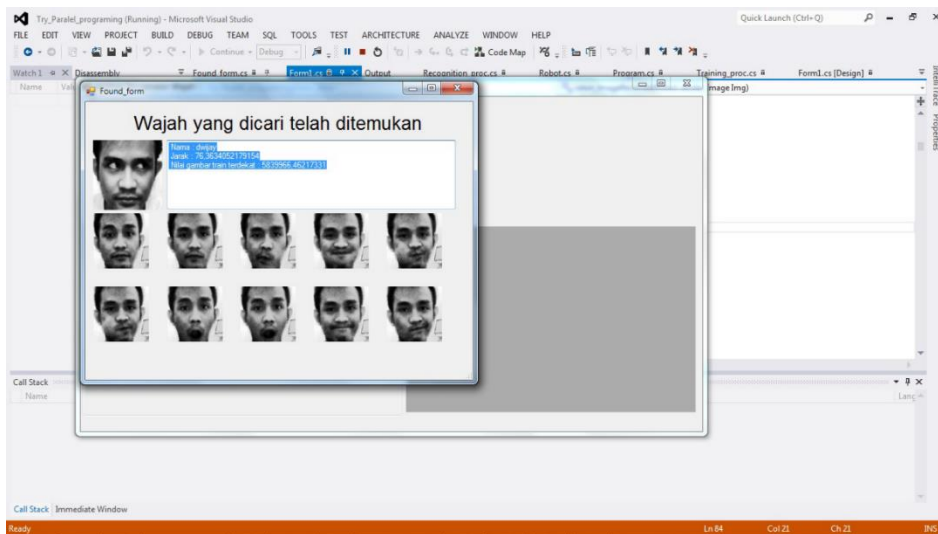
4. Menjalankan aplikasi program secara keseluruhan
5. Melakukan proses *training* terlebih dahulu pada beberapa orang
6. Menghubungkan Robotino sesuai IP yang ditentukan
7. Memberikan inputan pada program berupa nama orang yang akan dicari
8. Melihat pergerakan Robotino apakah sesuai dengan yang diinginkan dalam melakukan pencarian orang
9. Melihat pada layar PC, apakah citra *streaming* dari kamera Robotino telah ditampilkan pada layar PC
10. Melihat hasil pencarian apakah sudah sesuai atau belum

4.9.4. Hasil Pengujian

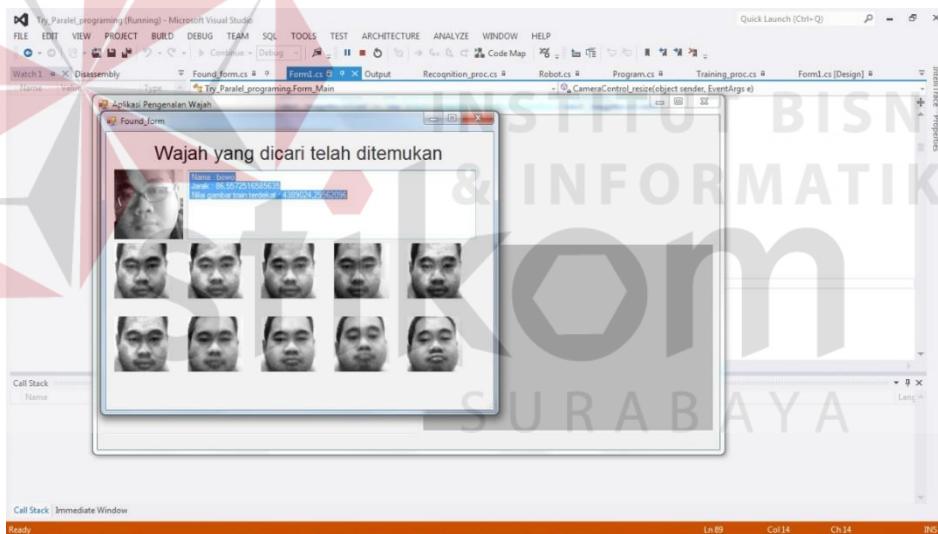
Setelah melalui seluruh prosedur pengujian diatas didapatkan hasil tingkat keberhasilan 100% dari 3 subjek uji. Dari ketiga wajah yang dideteksi menghasilkan hasil data yang sama dengan orang yang dicari, seperti pada gambar 4.11 sampai 4.13 berikut.



Gambar 4.13 Hasil pendeteksian wajah untuk orang pertama



Gambar 4.14 Hasil pendeteksian wajah untuk orang kedua



Gambar 4.15 Hasil pendeteksian wajah untuk orang ketiga

Dari hasil-hasil pengujian seperti pada Gambar 4.13 sampai dengan Gambar 4.15, Robotino akan mencari wajah sesuai dengan inputan dari *user*. Robotino yang telah berhasil menemukan wajah dari orang yang dicari akan menghasilkan *output* seperti pada Gambar 4.13 sampai 4.15.