

APLIKASI PENGENALAN WAJAH MENGGUNAKAN METODE *EIGENFACE* DAN JARAK *EUCLIDEAN*

Kresna Lita Maulana^{*)}, Achmad Hidayatno, dan Imam Santoso

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)E-mail: klita46@gmail.com}

Abstrak

Wajah merupakan salah satu identitas bagi setiap individu pada sistem biometrik. Wajah merupakan ciri unik dari setiap manusia yang dapat membedakan rupa antar manusia. Berbeda dengan manusia yang dapat mengenali wajah dengan mudah dan cepat, komputer tidak secepat dan semudah manusia. Pada komputer diperlukan suatu algoritme dalam pengenalan wajah. Pada Penelitian ini, dirancang suatu sistem pengenalan wajah menggunakan kamera *web* dan OpenCV yang terpasang pada Raspberry Pi 3 Model B. Masukan sistem berupa video *real-time* yang diperoleh dari kamera *web*. Metode yang digunakan pada pendeteksian wajah adalah metode Viola-Jones dan dalam pengenalan wajah digunakan metode *eigenface* dan jarak *euclidean*. Terdapat 5 responden yang diambil citra wajahnya sebagai *database*. Hasil yang diperoleh dari sistem ini adalah nama dari setiap responden yang terdapat pada *database*. Berdasarkan hasil pengujian pada kondisi dalam ruangan dihasilkan rata-rata akurasi sebesar 99,8%, sedangkan pada kondisi luar ruangan dihasilkan rata-rata akurasi sebesar 93,8%. Pada pengujian citra wajah yang diberi derau *salt & pepper* dengan kepadatan derau 0,001 dan 0,01 didapatkan bahwa program mampu mengenali wajah dengan benar. Program mampu mengenali wajah dengan benar pada citra yang dirotasi sebesar 10 derajat.

Kata kunci: Pengenalan wajah, algoritme eigenface, jarak euclidean, metode Viola Jones, OpenCV, Raspberry Pi.

Abstract

Facial recognition is one of the biometric systems. Face is the hallmark of every human being who can distinguish between human beings. Face recognition is widely used for identification systems on security systems and attendance machines. In this Research, designed a face recognition system using web camera and OpenCV mounted on Raspberry Pi 3 Model B. Input system of video obtained from camera. The method used on face detection is Viola-Jones method and in face recognition used *eigenface* and *euclidean* distance. Face image capture of 55 images from 5 people using web camera. The image is stored in a folder as a *database*. The result obtained from this system is the name of each respondent in the *database*. In the test also performed a system endurance test against *salt & pepper* and image rotation attacks. Based on the test result on indoor condition, the average value of accuracy is 99,8% and in outdoor condition, the average value of accuracy is 93,8%. Based on the image test of *salt & pepper* attack it was found that the program was able to correctly recognize the face at 0.001 and 0.01 noise density. Image testing against rotational attacks found that the program is only able to detect faces on rotated images of 10, 15, and 20 degrees. The program is not resistant to rotational attacks so it goes wrong recognizing the face.

Keywords: face recognition, eigenface algorithm, euclidean distance, Viola Jones method, OpenCV, Raspberry Pi.

1. Pendahuluan

Pengenalan wajah dalam beberapa tahun terakhir semakin berkembang dengan adanya kemajuan *software* maupun *hardware*, identifikasi seseorang berdasarkan biometrik telah berkembang dengan pesat. Metode pengenalan identitas seseorang yang banyak digunakan diantaranya berdasarkan nomor identitas unik seperti kartu identitas atau berdasar kan ingatan terhadap suatu sandi rahasia [14]. Metode tersebut banyak memiliki kekurangan seperti hilangnya kartu identitas atau lupa sandi dari ingatan seseorang. Ada dua jenis biometrik di antaranya adalah

physiological (iris mata, wajah dan sidik jari) dan *behavioural* (suara dan tulisan tangan) [13]. Pengenalan wajah merupakan suatu pengenalan pola (*pattern recognition*) yang khusus untuk kasus wajah. *Eigenface* adalah suatu metode pengenalan wajah yang berdsarkan pada algoritme *Principal Component Analysis* (PCA). *Principal Component Analysis* yang merupakan suatu metode ekstraksi ciri yang mampu mengidentifikasi ciri tertentu yang merupakan karakteristik suatu citra (dalam hal ini adalah wajah) [14]. Liton dan Abdulla [15] merancang sebuah program untuk mengenali wajah menggunakan algoritme *eigenface* dengan

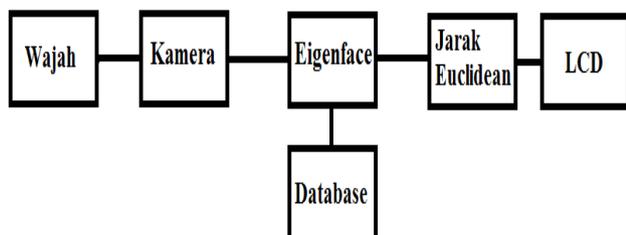
membandingkan citra wajah uji dan citra wajah pada *database* menggunakan matlab.

Pada Penelitian ini dirancang suatu sistem pengenalan wajah dengan menggunakan kamera *web* yang akan tersambung secara langsung dengan komputer mini Raspberry Pi sebagai perangkat pemroses. Sebelum melakukan pengenalan diperlukan pengambilan citra sebagai *database*. Citra pada *database* disimpan dengan format .JPEG. Masukan sistem merupakan video secara *real-time*. Metode pengenalan wajah yang digunakan adalah *Eigenface* dan jarak *Euclidean*. Selain itu, digunakan algoritme *haar cascade* yang berfungsi untuk mendeteksi wajah pada video. Hasil yang diperoleh merupakan nama dari seseorang yang sedang diuji secara *real-time*. Terdapat pengujian ketahanan sistem terhadap serangan-serangan *salt & pepper* dan rotasi. Inputan pada pengujian ketahanan adalah citra berformat .JPEG.

2. Metode

2.1. Blok Diagram Aplikasi Pengenalan Wajah

Sistem yang dirancang adalah sebuah program untuk mendeteksi dan mengenali wajah manusia. Jika terdeteksi adanya wajah manusia, maka sistem ini akan melanjutkan mengenali wajah yang direkam oleh kamera. Wajah akan dikenali berdasarkan pada nilai jarak antara wajah yang direkam oleh kamera dengan wajah di *database*. Sebelum melakukan pengenalan wajah diperlukan citra wajah yang digunakan sebagai *database* sehingga wajah bisa dikenali dengan benar. Pengambilan citra wajah sebagai *database* dilakukan menggunakan kamera *web*.



Gambar 1. Diagram blok sistem pengenalan wajah

2.2. Bagian Masukan

Pada bagaian ini kamera *web* memiliki dua fungsi sekaligus. Fungsi yang pertama, kamera dapat digunakan sebagai alat untuk mengambil citra wajah jika diinginkan untuk pengambilan citra sebagai *database*. Fungsi yang kedua, kamera digunakan sebagai alat merekam suatu video sebagai inputan. Pada pengenalan wajah, kamera akan merekam video secara *real-time* yang kemudian masukan dari kamera akan diproses oleh Raspberry Pi. Kamera *web* ini memiliki *port* keluaran berupa USB (*Universal Serial Bus*) yang disambungkan ke Raspberry Pi.

2.3. Bagian Pengolah

Raspberry Pi adalah komponen yang berfungsi sebagai pengolah. Raspberry Pi melakukan empat proses, yakni proses pengambilan citra wajah, *preprocessing*, deteksi letak wajah, dan pengenalan wajah. Pada saat melakukan pengenalan wajah peratama-tama video yang diambil secara *real-time* oleh kamera akan diperikasa apakah pada video terdapat wajah atau tidak. Jika terdapat objek yang merupakan wajah, maka objek tersebut akan langsung dilakukan proses pengenalan. Jika pada video tidak terdeteksi adanya wajah, maka sistem tidak akan melakukan pengenalan.

2.3.1. Proses Deteksi Wajah

Bagian ini berfungsi untuk mendeteksi wajah dari video yang diambil oleh kamera. Deteksi wajah ini menggunakan suatu *library* yang terdapat pada OpenCV, melalui API (*Application Programming Interface*) C++ bernama *cv2.CascadeClassifier*. Senarai tersebut berfungsi untuk mendeteksi objek yang direkam oleh kamera. Pada pendeteksian wajah maka digunakan *file XML* yang berisi data pengklasifikasi wajah tampak dari depan atau *frontal face*. Setelah mengetahui apa yang akan dideteksi maka dilakukan pendeteksian wajah menggunakan fungsi *library cv2.detectMultiScale*. Senarai ini berfungsi untuk mendeteksi wajah dari berbagai ukuran pada video.

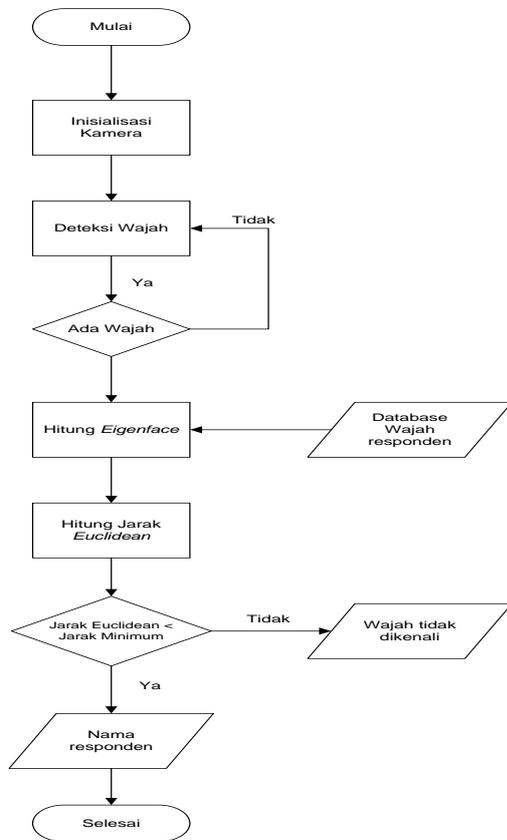
2.3.2. Proses Pengambilan Citra Wajah

Proses pengambilan citra wajah sebagai *database* akan dilakukan diawal sebelum pengujian dilakukan. Pengambilan citra dilakukan di dalam ruangan dengan pencahayaan yang cukup. Pada pengambilan citra wajah diambil sebanyak 11 citra wajah. Pada perancangan ini, citra wajah diambil menggunakan kamera *web*. Responden yang diambil pada perancangan ini ada 5 responden sehingga jumlah citra wajah untuk *database* ada sebanyak 55 citra wajah. Pada pengambilan citra wajah dilakukan pendeteksian wajah terlebih dahulu. Jika pada video terdapat wajah maka, program akan memotret wajah tersebut. Pada saat wajah dipotret oleh kamera dilakukan proses *preprocessing* pada citra tersebut. Tahapan dalam *preprocessing* di antaranya adalah mengubah ukuran jumlah piksel pada citra, normalisasi intensitas menggunakan perataan histogram, dan memotong citra wajah yang terdeteksi menjadi persegi.

2.3.3. Proses Pengenalan Wajah

Pengenalan wajah dilakukan setelah didapatkan *database* yang berisi citra wajah dari responden. Pada tahap pengenalan digunakan kamera *web* sebagai inputan. Terdapat tujuh *library* pada perancangan program pengenalan wajah, yaitu PIL, numpy, scipy, os, OpenCV,

sys, dan glob. Setiap library memiliki fungsi yang berbeda-beda.



Gambar 2. Diagram alir pengenalan wajah.

Secara matematis langkah pertama dalam pengenalan wajah adalah membuat suatu himpunan S yang terdiri dari citra-citra wajah sebagai *database*.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M\} \quad (1)$$

Setiap citra wajah pada *database* direpresentasikan sebagai $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M$ dengan M adalah jumlah citra wajah. Langkah kedua adalah mendapatkan nilai rata-rata dari citra wajah pada *database* dengan rumus:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (2)$$

Langkah ketiga adalah mencari selisih (Φ) antara citra di dalam *database* (Γ_i) dengan nilai rata-rata Ψ .

$$\Phi_i = \Gamma_i - \Psi \quad (3)$$

Langkah keempat adalah menghitung nilai matriks kovarian (C) karena secara matematis *principal component* dari suatu data adalah *eigenvector* dari matriks kovarian. *Principal Component Analysis* akan mencari sekumpulan vektor yang menggambarkan secara signifikan sebuah variasi data:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (4)$$

Dimana $A = [\Phi_1, \Phi_2, \dots, \Phi_M]$ dan A^T adalah matriks transpose A . Langkah kelima menghitung *eigenvalue* (λ) dan *eigenvector* (v) dari matriks kovarian (C).

$$(C - \lambda I) = 0 \text{ atau } (\lambda I - C) = 0 \quad (5)$$

Langkah keenam adalah mencari nilai *eigenface* (μ).

$$\mu_i = \sum_{k=1}^M v_{lk} \Phi_k \quad (6)$$

Dimana v adalah *eigenvector* dan Φ_k adalah selisih antara setiap training image Γ_i pada *database* dengan nilai rata-rata citra wajah Ψ . Sebuah citra wajah uji atau *test face* (Γ_{new}) akan dilakukan proses pengenalan. Citra wajah baru tersebut harus dilakukan proses perhitungan untuk mendapatkan *eigenface*. Setelah didapatkan nilai citra wajah baru atau *test face* barulah dilakukan pengenalan wajah dengan membandingkan nilai tersebut menggunakan jarak *euclidean*.

$$\mu_{new} = v(\Gamma_{new} - \Psi) \quad (7)$$

$$d(x, y) = \sqrt{\sum_{i=1}^M ((x_i) - (y_i))^2} \quad (8)$$

2.4. Bagian Keluaran

2.4.1. Layar LCD

Layar LCD berfungsi untuk menampilkan GUI (*Graphical User Interface*) dan melihat video secara *real-time*. Pada GUI tersebut, pengguna dapat memilih untuk mengambil data berupa citra wajah sebagai *database* atau menguji program untuk melakukan pengenalan wajah. Apabila pengguna ingin melakukan pengambilan citra wajah untuk disimpan sebagai *database* maka pengguna dapat mengklik *button* ambil citra



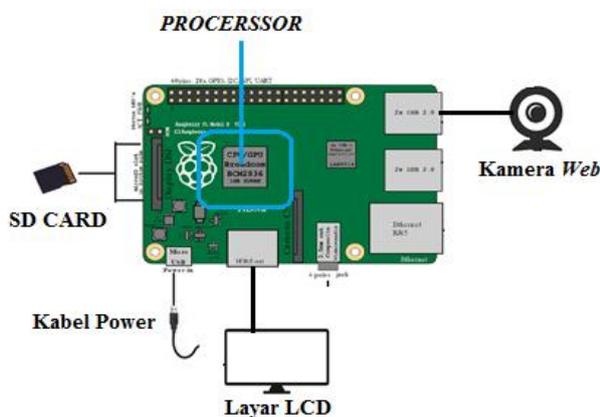
Gambar 3. Tampilan jenedela hasil pengenalan pada LCD.

2.5. Perancangan Perangkat Lunak

Perangkat lunak yang digunakan adalah Python versi 3.6 dan OpenCV versi 3.2. Program dibuat menggunakan aplikasi anaconda dan spyder 3.1.4. Setelah seluruh program selesai maka program hanya perlu dipindahkan ke Raspberry Pi. Python sudah terpasang secara langsung pada Raspberry Pi, sedangkan OpenCV harus diunduh dan dipasang terlebih dahulu pada Raspberry Pi. Sistem operasi yang digunakan pada Raspberry Pi adalah raspbian jessie. *Library* yang digunakan pada perancangan program di antaranya adalah OpenCV, numpy, scipy, tkinter, PIL, os, sys, runpy dan lain sebagainya. *Library* untuk membuat GUI adalah tkinter. Pada saat menjalankan program digunakan sebuah aplikasi yang bernama idle.

2.6. Perancangan Perangkat Keras

Perangkat keras yang digunakan dalam melakukan pengenalan wajah di antaranya adalah Layar LCD, Kamera web, Raspberry Pi, keyboard, mouse, kabel HDMI, dan Micro SD. Rangkaian keseluruhan alat ditunjukkan pada Gambar 4. Raspberry Pi diberi catu daya yang bersumber dari USB charger. Micro SD yang digunakan pada perancangan ini berkapasitas 16GB. Layar LCD yang digunakan pada perancangan ini berukuran 21 inch. Rangkaian ini juga digunakan untuk melakukan pengambilan citra wajah.



Gambar 4. Skematik raspberry Pi 3 Model B.

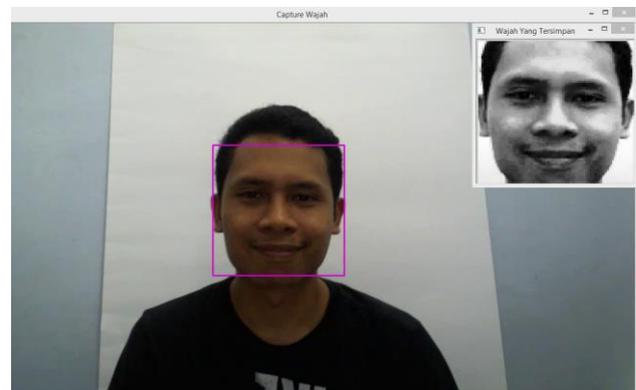
3. Hasil dan Analisa

Pada bab ini dibahas berbagai pengujian yang dilakukan pada program yang telah dirancang dan menganalisis hasil pengujian. Masukan yang digunakan adalah video *real-time*. Pengujian dilakukan di dalam dan luar ruangan. Pada saat di dalam ruangan digunakan pengujian menggunakan latar belakang karton berwarna putih. Pengujian di luar ruangan dilakukan pada saat siang hari. Posisi kamera hanya mengambil bagian wajah tampak depan saja. Sebelum pengujian, dilakukan pengambilan

citra wajah sebanyak 11 citra. Jumlah responden pada penelitian ini adalah 5 orang. Pengujian yang dilakukan dibagi menjadi lima bagian. pertama adalah pengujian pengambilan citra wajah, yaitu pengujian yang bertujuan untuk mengambil citra pada bagian wajah saja. Pada pengambilan citra wajah dilakukan proses *preprocessing* oleh program. Kedua adalah pengujian dalam ruangan, yaitu pengujian yang bertujuan untuk mengenali wajah seseorang di dalam ruangan. Hasil yang diperoleh dari pengenalan wajah kondisi di dalam ruangan akan dianalisis. Ketiga adalah pengujian luar ruangan, yaitu pengujian yang bertujuan untuk mengenali wajah seseorang di luar ruangan. Hasil yang diperoleh dari pengenalan wajah kondisi di luar ruangan akan dianalisis. Keempat adalah pengujian wajah dengan individu yang tidak ada di dalam *database*, yaitu pengujian pengenalan dengan menggunakan responden yang tidak ada di *database*. Pengujian wajah dengan citra yang diberi serangan-serangan, yaitu pengujian pengenalan dengan citra yang sudah diberi derau *salt & papper* dan rotasi citra.

3.1. Pengujian Pengambilan Citra Wajah

Pengujian pengambilan citra wajah bertujuan untuk mengambil citra wajah yang dideteksi oleh program kemudian di lakukan proses *preprocessing* dan menyimpan citra tersebut kedalam folder *database*.



Gambar 5. Pengujian pengambilan citra sebagai *database*

Tahap pertama sebelum dilakukan proses *preprocessing* adalah mengaktifkan kamera. Pada saat kamera aktif muncul sebuah jendela yang menampilkan video hasil rekaman kamera secara *real-time*. Tahap kedua adalah program akan mendeteksi keberadaan wajah. Pada saat terdapat wajah yang ditangkap kamera maka secara otomatis kamera memotret wajah tersebut dan dihasilkan citra wajah yang sudah dilakukan *preprocessing*. Proses yang dilakukan dalam *preprocessing* adalah mengubah ukuran citra, mengubah warna citra dari RGB menjadi aras keabuan. Pada saat citra berwarna sudah diubah menjadi aras keabuan dilakukan perataan histogram untuk mendapatkan citra dengan daerah tingkat keabuan yang

terdistribusi secara merata. Citra yang diambil oleh kamera disimpan di suatu folder dengan format .JPEG berukuran 100×100 piksel. Gambar 5, menunjukkan proses pengambilan citra wajah sebagai *database*.

3.2. Pengujian Pengenalan Kondisi Dalam Ruangan

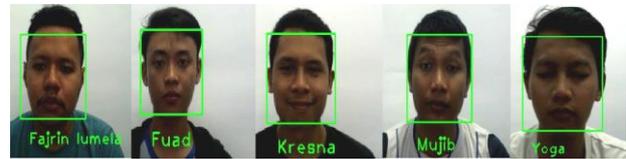
Pengujian pengenalan wajah di dalam ruangan menggunakan inputan yang berasal dari kamera *web* secara *real-time*. Resolusi yang digunakan adalah 640×480 piksel. Pengenalan wajah di dalam ruangan digunakan latar belakang karton yang berwarna putih. Pada saat kamera aktif maka program akan mendeteksi keberadaan wajah. Apabila terdapat wajah maka program akan menandai wajah tersebut dengan persegi berwarna hijau. Pada persegi hijau tersebut terdapat nama dari wajah yang dikenali. Pengujian dilakukan selama 10 detik dengan posisi wajah tampak dari depan. Pada saat pengujian akan terlihat nama dari wajah yang dikenali dan nilai jarak *euclidean* pada *python shell*. Nilai jarak *euclidean* akan berubah-ubah karna program terus menghitung secara *real-time*. Analisis yang dilakukan pada pengenalan wajah kondisi dalam ruangan adalah dengan membandingkan waktu pengujian dengan wajah yang dikenali. Video pengujian memiliki *frame rate* sebesar 15 *frame* per detik. Pengujian di luar ruangan memiliki jarak *euclidean* minimum sebesar 4500 yang diatur sebelum melakukan pengujian.

Tabel 1. Hasil pengujian akurasi pengenalan wajah kondisi dalam ruangan seluruh responden

| Responden | Jumlah orang | True | False | False | True | Akurasi |
|-----------|--------------|----------|----------|----------|----------|---------|
| | | Positive | Negative | Positive | Negative | |
| FL | 1 | 142 | 1 | 0 | 0 | 99% |
| KR | 1 | 143 | 0 | 0 | 0 | 100% |
| YG | 1 | 143 | 0 | 0 | 0 | 100% |
| FA | 1 | 143 | 0 | 0 | 0 | 100% |
| MJ | 1 | 143 | 0 | 0 | 0 | 100% |

Berdasarkan Tabel 1, setiap responden memiliki akurasi yang baik untuk kondisi dalam ruangan. Tingkat akurasi yang tinggi disebabkan intensitas cahaya pada saat pengambilan wajah sebagai *database* dan pada saat pengujian adalah sama. Pada saat pengujian setiap responden melakukan beberapa perubahan ekspresi wajah seperti tersenyum dan menaikkan alis. Perubahan ekspresi tidak berpengaruh pada pengenalan wajah untuk didalam ruangan dengan tingkat intensitas cahaya yang sama. Pada Tabel 4.2 akurasi tertinggi adalah 100% dan akurasi terendah adalah 99%. Nilai *true positive* responden KR, YG, FA, dan MJ adalah 143 yang artinya wajah dikenali dengan benar sebanyak 143 kali selama proses pengujian berlangsung. Nilai *false negative* pada responden FL adalah 1 yang artinya wajah salah dikenali sebanyak 1 kali selama proses pengujian berlangsung. Pada saat pengujian seluruh responden program berhasil melakukan

deteksi wajah dengan baik. Program tidak mendeteksi objek selain wajah sehingga nilai *true negative* adalah nol



Gambar 6. Hasil pengujian pengenalan wajah kondisi dalam ruangan yang dikenali dengan benar

3.3. Pengujian Pengenalan Kondisi Luar Ruangan Perindividu

Pengujian pengenalan kondisi luar ruangan merupakan pengujian yang dilakukan dengan memanfaatkan cahaya matahari. Pada pengujian ini kondisi luar ruangan sedang dalam keadaan cerah. Pengujian dilakukan pada waktu sore hari dengan tingkat intensitas cahaya matahari yang berubah-ubah. Pada pengujian ini akan dianalisis tingkat akurasi pengenalan setiap individu. Pengujian dilakukan selama 10 detik dengan posisi wajah tampak dari depan. Video pengujian memiliki *frame rate* sebesar 15 *frame* per detik. Rosulis video yang digunakan adalah 640×480 piksel. Pada pengujian ini setiap responden juga melakukan perubahan pada ekspresi wajah, seperti menaikkan alis, tersenyum, dan kaget. Kinerja program akan dilihat berdasarkan akurasi dari pengenalan wajah setiap individu. Pengujian di luar ruangan memiliki jarak *euclidean* minimum sebesar 5500 yang diatur sebelum melakukan pengujian.

Tabel 2. Hasil pengujian akurasi pengenalan wajah kondisi luar ruangan seluruh responden

| Responden | Jumlah orang | True | False | False | True | Akurasi |
|-----------|--------------|----------|----------|----------|----------|---------|
| | | Positive | Negative | Positive | Negative | |
| FL | 1 | 143 | 3 | 0 | 0 | 97% |
| KR | 1 | 125 | 18 | 0 | 0 | 87% |
| YG | 1 | 141 | 2 | 0 | 0 | 98% |
| FA | 1 | 125 | 18 | 0 | 0 | 87% |
| MJ | 1 | 143 | 0 | 0 | 0 | 100% |

Berdasarkan Tabel 2, nilai *true positive* pada responden MJ adalah 143 yang artinya wajah dikenali dengan benar sebanyak 143 kali selama proses pengujian berlangsung. Nilai *false negative* pada responden MJ adalah 0 yang artinya tidak ada wajah yang salah dikenali selama proses pengujian berlangsung. Akurasi terendah pada pengujian di luar ruangan adalah 87%. Penyebab dari kesalahan pengenalan adalah perubahan ekspresi wajah dari responden KR dan FA. Transisi perubahan ekspresi wajah dari kedua responden menjadi penyebab utama kesalahan dalam pengenalan. Jumlah *frame* dari setiap video adalah 143 *frame*. Pada saat pengujian program tidak mendeteksi objek selain wajah sehingga nilai dari *true negative* untuk

seluruh responden adalah nol. Gambar 8, menunjukkan kesalahan pada pengenalan wajah yang disebabkan apabila terjadi perubahan ekspresi pada wajah.



Gambar 7. Hasil pengujian pengenalan wajah kondisi luar ruangan yang dikenali dengan benar



Gambar 8. Hasil pengujian pengenalan wajah kondisi luar ruangan yang salah dikenali

3.4. Pengujian Pengenalan Wajah Individu yang Tidak Ada Di dalam Database

Pengujian ini dilakukan oleh tiga individu yang tidak terdapat pada database. Pada pengujian ini masukan berupa video yang sudah direkam menggunakan kamera. Jarak minimum pada pengujian kali ini adalah 4500. Apabila wajah memiliki nilai jarak lebih dari 4500 maka wajah akan dikategorikan sebagai ‘wajah tidak dikenali’. Video pada pengujian berdurasi 10 detik dengan frame rate sebesar 15 frame per detik



Gambar 9. Hasil pengujian akurasi wajah pada individu yang tidak ada di dalam database

Tabel 3. Hasil pengujian akurasi wajah pada individu yang tidak ada di dalam database

| Nama responden | Jumlah orang | True Positive | False Negative | False Positive | True Negative | Akurasi |
|----------------|--------------|---------------|----------------|----------------|---------------|---------|
| IK | 1 | 143 | 0 | 0 | 0 | 100% |
| RQ | 1 | 141 | 2 | 0 | 0 | 98% |
| RZ | 1 | 143 | 0 | 0 | 0 | 100% |

Tabel 3, menunjukkan bahwa ketiga responden tersebut berhasil tidak dikenali karena wajah ketiga responden tidak terdapat pada database. Responden RQ sempat dikenali sebagai responden YG pada detik kedelapan pada saat ekspresi wajah dari responden RQ menaikkan alisnya.

3.5. Pengujian Pengenalan Wajah yang Diberi Derau Salt & Pepper

Pada pengujian ini responden yang digunakan adalah sebanyak lima. Derau salt & pepper berupa derau yang menyerupai titik-titik garam dan merica. Pada citra warna RGB, maka deraunya berupa titik-titik dengan warna red, green, dan blue. Kepadatan derau yang digunakan pada percobaan ini adalah 0,001, 0,01, dan 0,1.



Gambar 10. Hasil pengujian pengenalan wajah dengan derau salt & pepper (0,001, 0,01, 0,1) pada kondisi dalam dan luar ruangan

Gambar 10, memperlihatkan hasil pengenalan pada citra berderau salt & pepper. Pada saat nilai kepadatan derau adalah 0,001 dan 0,01 didapatkan bahwa program mampu mengenali kelima responden dengan benar. Pada pengujian ini dengan kepadatan derau 0,1 program hanya mampu mengenali tiga responden.

3.6. Pengujian Pengenalan Wajah Pada Citra yang Dirotasi

Pada pengujian dengan serangan rotasi citra, citra wajah akan dirotasi dengan 4 sudut kemiringan yaitu 10, 15, 20 dan 25 derajat. Pada pengujian ini akan dilihat kinerja sistem dalam mendeteksi dan mengenali wajah pada sebuah citra.



Gambar 11. Hasil pengujian pengenalan wajah dengan serangan rotasi citra responden fajrin

Gambar 11, memperlihatkan bahwa serangan rotasi citra mempengaruhi hasil pengenalan wajah. Tidak hanya mempengaruhi hasil dari pengenalan wajah, citra yang dirotasi juga menyebabkan algoritme *haar cascade* tidak bekerja dengan baik. Hasil pengenalan wajah dengan sudut kemiringan 10 derajat untuk kondisi dalam ruangan memiliki akurasi 100%. Pada saat sudut kemiringan 20 dan 25 derajat algoritme *haar cascade* tidak dapat mendeteksi adanya wajah pada citra. Pada kondisi luar ruangan algoritme *haar cascade* mampu mendeteksi wajah pada sudut kemiringan 10, 15, dan 20 derajat. Program mengalami salah mengenali pada kondisi luar ruangan dengan sudut kemiringan 10, 15, dan 20 derajat. Pada sudut 25 derajat algoritme *haar cascade* tidak mampu mendeteksi adanya wajah sehingga batas sudut yang dapat di deteksi dengan menggunakan algoritme *haar cascade* adalah 20 derajat.

4. Kesimpulan

Pada pengenalan wajah kondisi dalam ruangan diperoleh akurasi sebesar 99% dan 100%. Pada pengenalan wajah kondisi luar ruangan diperoleh akurasi tertinggi sebesar 100% untuk responden MJ, sedangkan hasil terendah adalah 87% untuk responden KR dan FA. Program berhasil tidak mengenali wajah pada pengujian individu

yang tidak ada di dalam *database*. Pengujian pengenalan wajah untuk citra berderau *salt & pepper* dengan nilai kepadatan 0,001 dan 0,01 didapatkan bahwa program mampu mengenali kelima responden dengan benar. Pada pengujian dengan kepadatan derau 0,1 program hanya mampu mengenali tiga responden. Pada pengujian rotasi citra dengan sudut kemiringan 10 derajat pada kondisi dalam ruangan didapatkan bahwa program mampu mengenali kelima responden dengan benar, sedangkan saat sudut kemiringan 20 dan 25 derajat algoritme *Haar cascade* tidak dapat mendeteksi adanya wajah pada citra. Penelitian ini dapat dikembangkan lebih lanjut dengan membandingkan penelitian ini dengan metode pengenalan wajah yang lain seperti LBPH, *fisherface*, LDA, dan lain sebagainya agar diketahui metode terbaik dalam pengenalan wajah. Pada penelitian selanjutnya dapat menambahkan variasi ekspresi dan posisi wajah pada pengujian pengenalan.

Referensi

- [1] P. E. Diah. "Pengenalan Wajah Menggunakan Metode *Principal Component Analysis* (PCA) Untuk Aplikasi Sistem Keamanan Rumah," 2013.
- [2] Immanuela P. Saputro, Ernawati, dan B. Yudi Dwiandiyanta, "Pengenalan Ekspresi Wajah Menggunakan *Wavelet* Gabor dan *Backpropagation*," *Jurnal Elektro*, vol. 8, no. 2, Oktober, 2015.
- [3] Febrinata, Kaziz, "Simulasi dan Analisis *Multiple Object Tracking* Berbasis Citra dengan Metode *Hierarchical Particle Filter*," Tugas Akhir, Telkom University, 2013.
- [4] Putra, Darma, *Pengolahan Citra Digital*. Depok: Andi Offset, 2010
- [5] S. N. Candra. "Mengubah Citra Berwarna Menjadi *Gray-Scale* dan Citra Biner," *Jurnal Teknologi Informasi Dinamik*, vol. 16, no. 1, Januari, 2011.
- [6] A. F. Riza, "Pengenalan Pola Tanda Tangan dengan Menggunakan Metode *Principal Component Analysis* (PCA)," Tugas Akhir, Universitas Dian Nuswantoro, 2013.
- [7] H. K. Adrian "Perancangan Sistem Pengenal Garis Utama Telapak Tangan Pada Sistem Presensi Menggunakan Metode *Principal Component Analysis* (PCA) dan Jarak *Euclidean*," *Jurnal Transient*, vol. 4, no. 1, Maret, 2015.
- [8] M. F. Eko. "Implementasi Metode *Median Filter* dan *Histogram Equalization* Untuk Perbaikan Citra Digital," *Jurnal Pelita Informatika*, vol. 16, no. 3, Juli, 2017.
- [9] Harris Simaremare dan Agung Kurniawan. "Perbandingan Akurasi Pengenalan Wajah Menggunakan Metode LBPH dan *Eigenface* dalam Mengenali Tiga Wajah Sekaligus secara *Real-Time*," *Jurnal Sains, Teknologi dan Industri*, vol. 14, no. 1, Desember, 2016.
- [10] Dulal Chakraborty, Sanjit Kumar Saha, dan Md. Al-Amin Bhuiyan, "Face Recognition using *Eigenvector* and *Principle Component Analysis*," *international Journal of Computer Applications*, vol. 50, no. 10, Juli, 2012.

- [11] Y. S. Nimas, "Perbandingan Ukuran Jarak Pada Proses Pengenalan Wajah Berbasis *Principal Component Analysis* (PCA)," Tugas Akhir, Institut Teknologi Sepuluh Nopember, 2011.
- [12] P. Viola dan M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Conf. Comput. Vis. PATTERN Recognit.*, 2001.
- [13] S. Y. Bertua Frans. "Sistem Pengenalan Wajah Menggunakan Metode *Principal Component Analysis* (PCA) dan Jaringan Syaraf Tiruan," *Jurnal Transmisi*, vol. 15, no. 3, 2013.
- [14] Wicaksono, Galih. "Sistem Identifikasi Garis Utama Telapak Tangan Menggunakan Metode *Principal Component Analysis* (PCA) dan Jarak *Euclidean*," *Jurnal Transient*, vol. 3, no. 1, Maret, 2014.
- [15] P.C Liton dan S.A Abdulla, "*Face Recognition Using Principal Component Analysis Method*," *International Journal of Advance Research in computer Engineering & Technology*, vol. 1, November, 2012.