

Face Recognition Berbasis Raspberry Pi Pada Keamanan Pintu Otomatis

Mauludi Manfaluthy, S.T., M.T.¹⁾, Sinka Wilyanti, S.T., M.T.²⁾, & Yunan Lasito³⁾

¹⁾²⁾³⁾ Fakultas Teknologi dan Ilmu Komputer Jurusan Teknik Elektro
Institut Teknologi dan Kesehatan Jakarta

JL. Jatiwaringin Raya No. 278, Pondok Gede 17411, Jakarta

Website: <https://itkj.ac.id>, E-mail: mauludi@itkj.ac.id, sinka@itkj.ac.id, yunan.lasito@itkj.ac.id

Abstrak – Dengan mulainya revolusi industri 4.0, dimulailah tren dunia industri yang menggabungkan teknologi otomasi dengan teknologi cyber. Untuk itu diperlukan sistem keamanan sebagai respon semakin berkembangnya teknologi cyber ini. Salah satu solusi keamanan dalam melakukan otentikasi adalah menggunakan bagian tubuh manusia yaitu wajah. Sistem dapat mendeteksi objek wajah sebagai citra masukan dari kamera. Setelah objek terdeteksi, sistem akan melakukan pencocokan wajah dengan citra wajah yang terdapat pada database sistem. Sistem ini merupakan penerapan Computer Vision dalam sistem keamanan. Citra akan diproses dengan menggunakan metode Haar Cascade untuk mendeteksi obyek wajah yang terdapat pada citra. Selanjutnya dengan metode Eigenface untuk membandingkan wajah terdeteksi dengan wajah pada database. Kedua metode ini akan diproses menggunakan Raspberry Pi. Tiga user melakukan uji sistem dengan kondisi yang berbeda-beda. Untuk user yang sudah terdaftar dalam database, kunci pintu akan terbuka otomatis. Sistem keamanan buka pintu akan menolak user yang tidak tersimpan dalam database dan mengirimkan notifikasi melalui aplikasi Telegram dan Whatsapp. Rata-rata akurasi pengenalan wajah berkisar 80% - 90% dengan jarak kamera sejauh 0.45 m pada ruangan dengan cahaya yang baik. Akurasi dan kemampuan respon sangat dipengaruhi oleh jarak, spesifikasi kamera, sudut, dan intensitas cahaya.

Abstract – The 4.0 industrial revolution, industry trends began to combine automation technology with cyber technology. For this reason, a security system is needed as a response to the growing development of this cyber technology. One of the security solutions in doing authentication is to use a part of the human body, the face. The system can detect face objects as input images from the camera. After the object is detected, the system will do a matching face with the face image contained in the system database. This system is using the application of Computer Vision in the security system. The image will be processed using the Haar Cascade method to detect facial objects contained in the image. The Eigenface method to compare detected faces with faces in the database. Both of these methods will be processed using Raspberry Pi. Three users test the system with different conditions. For users already registered in the database, the door lock will open automatically. The open door security system will reject users that are not stored in the database and send notifications via the Telegram and Whatsapp applications. The average accuracy of face recognition ranges from 80% - 90% with the distance of the camera as far as 0.45 m in a room with good light. Accuracy and response capabilities are greatly influenced by distance, camera specifications, angle, and light intensity.

Keywords: Face Recognition, Eigenface, Raspberry Pi, Solenoid Lock.

1 Pendahuluan

Pemanfaatan teknologi biometrik sangatlah handal dalam membedakan identitas manusia satu dan lainnya. Penggunaan ciri khas panca indera manusia untuk dengan

memanfaatkan sidik jari, sidik mata, suara, wajah dapat diterapkan dalam perancangan sistem keamanan. Dalam penelitian ini penulis menggunakan pengenalan wajah (face recognition) untuk sistem keamanan kunci rumah.

Penelitian ini di latar balakangi oleh semakin rawannya tingkat keamanan dan keunggulan panca indera manusia khususnya wajah yang sulit untuk diduplikasi. Pada prinsipnya terlebih dahulu direkam gambar/wajah dari orang-orang yang memiliki hak akses penuh terhadap rumah yang akan dipasang kunci.

Dalam penelitian ini digunakan metode Haar Cascade dalam mengidentifikasi atau melakukan klasifikasi wajah melalui kamera Pi, selanjutnya disimpan dalam database. Dengan metoda Eigenface dicocokkan obyek wajah yang terdeteksi terhadap wajah pada basis data, proses ini dilakukan dengan bantuan Raspberry Pi serta solenoid door lock sebagai output untuk gerakan membuka/menutup pintu, selain itu juga dikirimkan notifikasi lewat aplikasi Telegram dan Whatsapp kepada pemilik rumah yang berisikan adanya seseorang yang mencoba membuka pintu rumahnya.

2 Landasan Teori

2.1 Computer Vision

Computer vision adalah suatu ilmu yang mempelajari proses rekonstruksi, penafsiran, dan pemahaman tampilan 3-dimensi dari tampilan 2- dimensinya. *Computer vision* menangani pemodelan dan meniru wawasan manusia dengan bantuan komputer. Untuk memahami dan menyimulasikan pengoperasian sistem penglihatan manusia, computer vision berdasar pada penggabungan pengetahuan dari ilmu komputer, elektro, matematika, fisiologi, biologi dan ilmu kognitif[1]. Fungsi-fungsi pendukung yang diperlukan untuk mendekati kemampuan manusia dalam menangkap informasi, sebuah teknologi secara penuh antara lain:

a. Proses pengumpulan citra/gambar (*image acquisition*). Pemerolehan citra pada manusia bermula dengan mata, maka informasi visual diterjemahkan ke dalam format yang kemudian akan dimanipulasi oleh otak. Seperti proses tersebut di atas, computer vision memerlukan mata untuk menangkap isyarat visual. Umumnya mata pada computer vision adalah kamera video. Kamera akan menterjemahkan adegan atau citra. Output kamera adalah isyarat analog, di mana frekuensi dan Amplitudonya Merepresentasikan detail daripada ketajaman di tempat kejadian (frekuensi berkaitan dengan sinyal dalam satu saat, manakala amplitudo dikaitkan dengan sinyal elektrik yang tinggi yang dihasilkan. Kamera memperhatikan kejadian pada satu pada satu masa, mengimbas dan menyaring ke dalam ratusan garisan mendatar yang sama. Setiap baris mewujudkan sinyal analog yang Amplitudonya menguraikan perubahan kecerahan (*brightness*) di sepanjang garisan sinyal. Untuk itu diperlukan Analog to Digital Converter untuk memprosesnya. ADC ini akan mengubah sinyal analog yang diwakili dalam bentuk sinyal informasi tunggal ke dalam aliran beberapa binary. Binary ini kemudian disimpan dalam memori dan akan menjadi data mentah (*raw data*) yang akan diproses [2].

b. Pengolahan Citra.

Dalam proses ini *computer vision* akan melibatkan beberapa manipulasi utama (manipulasi awal) dari data binari yang dihasilkan dalam proses pemerolehan citra. Pemrosesan citra membantu untuk memperbaiki dan meningkatkan kualitas citra itu sendiri, agar dapat dianalisis dan diproses dengan lebih efisien. Pemrosesan citra akan meningkatkan perbandingan isyarat kepada bunyi bising (*signal to noise ration*). Isyarat ini adalah informasi yang akan merepresentasikan objek dalam citra.

c. Analisis data citra

Pada peringkat ini scene akan dieksplorasi ke dalam bentuk ciri utama objek melalui proses investigasi. Komputer akan mulai melihat bilangan biner yang mewakili informasi visual untuk mengenal pasti ciri ciri tertentu dan karakteristik citra. Dalam proses yang lebih khusus, program analisis citra digunakan untuk mencari sepadan objek dalam citra. Kelebihan terbentuk antara objek dan latar belakang atau antara dua objek tertentu. Kelebihan ini akan dikesan sebagai hasil daripada perbezaan dalam tahap kecerahan (*brightness*) pada sisi yang berbeda dengan salah satu batasnya.

d. Proses memahami data citra.

Proses ini adalah langkah terakhir dalam proses *computer vision*, di mana objek tertentu dan hubungan mereka dikenal secara pasti. Bagian ini akan melibatkan kecerdasan buatan. Pemahaman berkaitan dengan template yang sepadan dengan wujud dalam *scene*. Fungsi ini menggunakan program pencarian (*search programme*) dan teknik pencocokan pola (*pattern matching*).

2.2 Pustaka OpenCV

Open CV adalah sebuah perangkat lunak yang bertujuan untuk pemrosesan citra dinamik dalam waktu nyata, yang dibuat oleh Intel Corporation, Program ini adalah *open source*. Program ini didedikasikan kebanyakannya untuk pemrosesan citra real time. Jika ditemukan *Perfomanced primitif Integrated* dari Intel dalam sistem komputer, maka akan mempercepat proses kerja program secara automatik [3].

2.3 Eigenfaces

Untuk mengurai informasi yang relevan dari sebuah pola wajah maka diperlukan metode *eigenface*, metoda ini akan mengubahnya pada satu set kode paling efisien dan akan dibandingkan terhadap database dari beragam wajah yang telah dikodekan sebelumnya[4].

Proses *eigenface* ini menggunakan rumus matematika untuk mendapat karakteristik dari suatu citra.

Ada tiga fase yang dharus dilalui,

- Fase pengambilan citra yang akan digunakan sebagai *training set*. Hal ini diperoleh melalui *input* berupa kamera Pi.
- Fase *training*. Pada fase ini, citra yang telah diperoleh digunakan sebagai *training set*. [5] Dimensi, orientasi dan intensitas cahaya harus disamakan untuk semua citra. Setiap citra direpresentasikan oleh sebuah vektor

$M \times M$, dimana M adalah dimensi citra dalam ukuran piksel.

- c. Fase pengenalan. Sebuah citra baru akan didapatkan kemudian setelah citra tersebut dikurangi dengan citra rata rata. Hasil perhitungan dengan menggunakan *Principal Component Analysis* (PCA).

2.4 Algoritma Eigenfaces

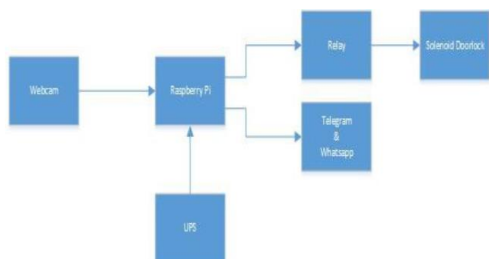
Untuk menghasilkan *eigenface*, sekumpulan citra digital dari wajah manusia diambil pada kondisi pencahayaan yang sama kemudian diproses pada resolusi yang sama (misal $a \times b$), kemudian citra tadi diperlakukan sebagai vektor dimensi $a \times b$ dimana komponennya diambil dari nilai piksel citra [6].

Algoritma *eigenface* secara keseluruhan cukup sederhana. *Image* matriks (Γ) direpresentasikan ke dalam sebuah himpunan matriks ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$). Cari nilai rata-rata (Ψ) dan gunakan untuk mengekstraksi *eigenvector* (v) dan *eigenvalue* (λ) dari himpunan matriks. Gunakan nilai *eigenvector* untuk mendapatkan nilai *eigenface* dari citra. Apabila ada sebuah citra baru atau *test face* (Γ_{new}) yang ingin dikenali, proses yang sama juga diberlakukan untuk citra (Γ_{new}), untuk meng-ekstraksi *eigenvector* (v) dan *eigenvalue* (λ), kemudian cari nilai *eigenface* dari citra *test face* (Γ_{new}) [7].

3 Perancangan Sistem

Penelitian ini menerapkan metode pengenalan wajah (*face recognition eigenface*) untuk meningkatkan keamanan pada pintu. Proses training dan testing metode *eigenface* menggunakan Python terhubung dengan Raspberry Pi. Berdasarkan hasil *training* dan *testing*, solenoid akan membuka dan mengunci pintu rumah. Kendali solenoid menggunakan Raspberry Pi yang mendapatkan masukan dari proses *training* dan *testing eigenface*. Jika *testing eigenface* bersesuaian terhadap *training*, maka Raspberry Pi seperti pada blok diagram (gambar 1) mengirimkan data untuk proses pembukaan pintu dan begitu juga sebaliknya jika hasil testing *unknown* maka Raspberry Pi mengirimkan notifikasi pesan melalui Telegram dan pintu tetap terkunci.

Waktu penelitian dilakukan selama 3 bulan dari bulan April s/d Juni 2019. Penelitian ini terdiri dari lima tahap yaitu: studi literatur, perancangan alat, pembuatan alat, pengujian dan dokumentasi.



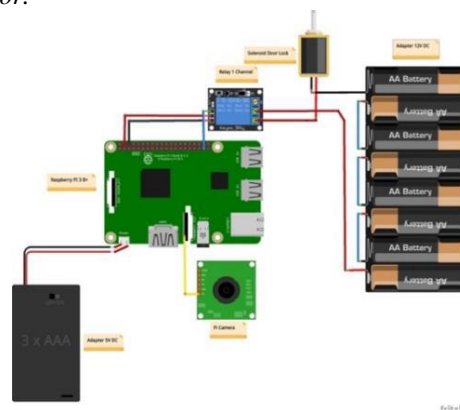
Gambar 1. Blok Diagram Sistem

Dari gambar 1, dapat dijelaskan prinsip kerja dari sistem ini. Pi Camera akan mendeteksi orang dengan cara mencari wajah dari orang tersebut lalu membuat sebuah kotak yang menandakan wajahnya. Ketika terdeteksi wajah orang,

Raspberry Pi akan melakukan proses pengambilan citra dari camera. Citra yang diambil kemudian dilakukan proses *cropping* atau pemotongan gambar dengan hanya mengambil bagian kotak yang menandakan wajahnya. Selanjutnya citra yang diambil nantinya akan diproses untuk mencari identifikasi (ID) untuk kemudian dilakukan pencocokan dengan database yang telah berisikan ID-ID pengguna yang sudah didaftarkan. Apabila citra yang diproses mendapatkan kecocokan dengan database maka Raspberry Pi akan mengirimkan sinyal digital "1" yang membuat relay keadaan *close* atau tertutup dan membuat *solenoid door lock* dalam kondisi membuka pintu. Namun apabila citra yang diproses tidak menemukan kecocokan dengan database, maka Raspberry Pi akan mengirimkan pemberitahuan dan mengirimkan citra tersebut ke aplikasi telegram dan whatsapp pemilik rumah.

3.1 Perancangan Perangkat Keras

Raspberry Pi terhubung ke kamera Pi yang ditunjukkan pada gambar 2. Raspberry Pi memiliki kemampuan pemrosesan yang sama dari prosesor inti tunggal dengan grafis bawaan yang dapat mendukung standar video hingga 1080p menggunakan *port High Definition Media Input* (HDMI). Sirkuit relay terhubung ke Raspberry Pi dan menyediakan antarmuka ke *solenoid door lock*. Statusnya dapat ditunjukkan dengan menggunakan indikator LED dan / atau selanjutnya mengirimkan pesan notifikasi melalui aplikasi telegram dan whatsapp menggunakan koneksi internet jika wajah tidak dikenali. Terakhir, daya keseluruhan yang dibutuhkan kurang dari 1 A, sehingga catu daya yang ringkas, seperti pengisi daya ponsel cerdas atau *powerbank* dapat digunakan sebagai catu daya Raspberry Pi dan catu daya 12 V untuk operasi *solenoid lock door*.



Gambar 2. Skematik Sistem Elektronik

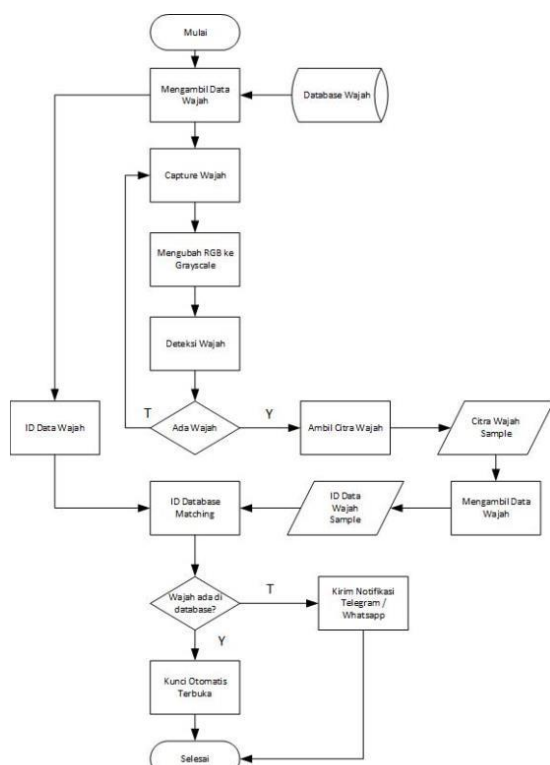
Dalam penelitian ini, port *input* eksternal terpasang dengan kamera Pi yang terhubung melalui slot kamera pada Raspberry Pi. Kamera ini memiliki spesifikasi resolusi 5MP dan mampu memberikan kecepatan data gambar dan video yang baik.

Modul relay yang digunakan adalah modul dengan 1 kanal, berfungsi untuk pemutus otomatis tegangan pada sistem. Pemilihan modul relay karena telah memiliki pin *power*, *ground* dan *input* dimana *power* dan *ground* sebagai sumber tegangan dan *grounding*, pin *input* disambungkan ke pin GPIO 14 pada Raspberry Pi 3 sebagai pengatur

keadaan relay. *Solenoid door lock* digunakan sebagai aktuator sehingga nantinya pintu pada sistem ini dapat terbuka secara otomatis sesuai dengan keadaan relay. *Solenoid door lock* disambungkan ke blok terminal beban pada relay dengan keadaan *normaly open*. *Solenoid door lock* yang digunakan pada sistem ini digerakkan dengan tegangan 12V DC sehingga untuk sumber salah satu kaki dihubungkan pada catu daya dan satu kaki dihubungkan dengan relay dimana dipasangkan ke *ground* Raspberry Pi 3.

3.2 Perancangan Perangkat Lunak

Setelah proses perancangan sistem secara perangkat keras, dilakukan tahap perancangan perangkat lunak sistem. Perangkat lunak adalah istilah umum untuk data yang diformat dan disimpan secara digital, berbeda dengan perangkat keras karena merupakan bagian sistem yang tidak berwujud. Perancangan ini terdiri atas perancangan perangkat lunak pada Raspberry Pi dengan menggunakan bahasa pemrograman Python dan perancangan perangkat lunak pada aplikasi Telegram dan Whatsapp. Pemrograman dilakukan berdasarkan pada *flowchart* sistem seperti yang terlihat pada gambar 3.



Gambar 3. Flowchart Sistem

Ketika alat mulai dinyalakan, maka akan dilakukan terlebih dahulu pencarian ciri khusus pada citra database. Ciri yang diambil adalah ID dari database wajah. Nantinya data ID dari masing-masing database disimpan pada library Python. Setelah itu maka Raspberry Pi akan membuat kamera melakukan *capture* wajah dan akan terus melakukan proses mendeteksi sebuah wajah. Apabila sistem mendeteksi adanya wajah manusia, maka kamera akan mengambil citra hanya pada bagian wajah (terjadi proses *cropping*). Citra

wajah yang didapatkan kemudian dilakukan proses pencarian ID. Data ID dari citra wajah yang diambil kemudian akan dilakukan proses pencocokan ciri (*ID Database Matching*) dengan data database wajah yang telah disimpan sebelumnya. Apabila citra dikenali dengan salah satu citra wajah yang ada pada database, maka sistem akan membuat relay ke keadaan *enable*. Relay yang dalam kondisi *close* akan membuat *solenoid lock door* mendapatkan tegangan sehingga pintu terbuka. Sedangkan apabila citra yang diambil tidak dikenali, maka sistem akan mengirimkan pesan notifikasi melalui internet ke Telegram dan Whatsapp yang berisi citra wajah yang tidak dikenali tersebut.

3.3 Perhitungan Eigenface

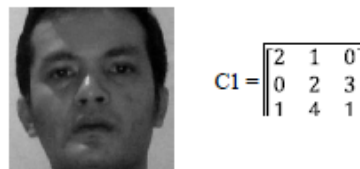
Berikut adalah tahapan pengenalan wajah:

1. Membuat suatu himpunan matriks dari *database*,
2. Cari nilai tengah atau *mean*,
3. Hitung perbedaan/selisih antara *training image* dengan nilai tengah,
4. Lakukan perhitungan nilai matriks kovarian,
5. Cari *eigenvalue* dan *eigenvector*,
6. Tentukan nilai *eigenface*,
7. Identifikasi.

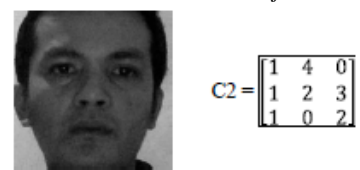
Sub-bab berikut ini akan menjelaskan tahapan-tahapan diatas.

3.3.1 Flatvector Matriks Citra

Tahapan awal adalah membuat himpunan S matriks yang terdiri dari seluruh training image ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$). Yang diambil sample data wajah seperti terlihat pada gambar 4 dan 5 yang memiliki nilai matriks C1 dan C2.



Gambar 4. Citra wajah 1



Gambar 5. Citra wajah 2

3.3.2 Perhitungan Mean (Ψ)

Untuk mencari nilai tengah atau *mean* (Ψ). Maka jumlahkan matrik citra wajah 1 dan wajah 2, selanjutnya dibagi dengan jumlah data wajah yang ada di database.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma$$

$$\Psi = \frac{1}{2} \sum_{n=1}^2 \Gamma = \frac{1}{2} \left(\begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 3 \\ 1 & 4 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 4 & 0 \\ 1 & 2 & 3 \\ 1 & 0 & 2 \end{pmatrix} \right)$$

$$\Psi = \begin{pmatrix} 1.5 & 2.5 & 0 \\ 0.5 & 2 & 3 \\ 1 & 2 & 1.5 \end{pmatrix}$$

3.3.3 Perbedaan Nilai Training Image dengan Nilai Tengah

Tahapan selanjutnya adalah mencari selisih (ϕ) antara *training image* (Γ) dengan nilai tengah (Ψ) dengan memakai nilai tengah citra diatas. Dengan jalan mengurangi *training image* (Γ) terhadap nilai tengah (Ψ).

$$\begin{aligned} \phi_i &= \Gamma_i - \Psi \\ \phi_1 &= \Gamma_1 - \Psi = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 3 \\ 1 & 4 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \\ \phi_2 &= \Gamma_2 - \Psi = \begin{bmatrix} 1 & 4 & 0 \\ 1 & 2 & 3 \\ 1 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \end{aligned}$$

3.3.4 Perhitungan Matriks Kovarian

Untuk menghitung *eigenvalue* (λ) dan *eigenvector* (v) digunakan Matriks kovarian (C), seperti berikut ini

$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T \\ L &= A^T A \quad L = \phi_n \phi_n^T \\ L &= \begin{bmatrix} 1 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 1 \end{bmatrix} X \\ L &= \begin{bmatrix} 6 & 0 & 6 \\ 0 & 1 & 0 \\ 6 & 0 & 9 \end{bmatrix} \end{aligned}$$

3.3.5 Perhitungan Eigenvalue dan Eigenvector

Untuk menghitung *eigenvalue* (λ) dan *eigenvector* (v) matriks kovarian (C), melalui rumus berikut ini:

$$\begin{aligned} Lxv &= \lambda xv \\ Lxv &= \lambda xv \\ L - \lambda I &= 0 \text{ atau } \lambda I - L = 0 \\ 0 &= \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 6 & 0 & 6 \\ 0 & 1 & 0 \\ 6 & 0 & 9 \end{bmatrix} \\ 0 &= \det \begin{bmatrix} \lambda - 6 & 0 & 6 \\ 0 & \lambda - 1 & 0 \\ 6 & 0 & \lambda - 9 \end{bmatrix} \end{aligned}$$

eigenvalue yang didapat adalah $\lambda_1 = 1, \lambda_2 = 1.3153, \lambda_3 = 13.6847$

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Untuk mendapatkan *Eigenvector* (v) dengan jalan substitusi *eigenvalue* (λ) kepersamaan $(\lambda - L)v = 0$.

Setelah *Eigenvector* dari setiap *eigenvalue* didapat lalu dihimpun menjadi satu matriks. Maka *eigenvector* matriks L adalah

$$\begin{bmatrix} 0 & 0.7882 & 0.6154 \\ -1 & 0 & 0 \\ 0 & -0.6154 & 0.7882 \end{bmatrix}$$

3.3.6 Perhitungan Eigenface

Setelah mendapatkan *eigenvector* (v), bisa didapatkan perhitungan *eigenface* (μ) melalui rumus berikut ini:

$$\mu_i = \left[\frac{1}{M} \sum_{n=1}^M v_{ik} \phi_k \right], i = 1, \dots, M$$

Cari nilai *eigenface* (μ):

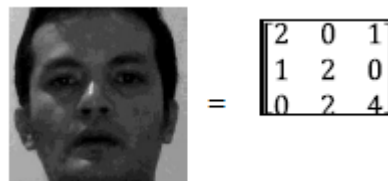
$$\begin{aligned} \mu_i &= \frac{1}{M} \sum_{n=1}^M \\ \mu_1 &= v \times \phi_1 \\ &= \begin{bmatrix} 0 & 0.7882 & 0.6154 \\ -1 & 0 & 0 \\ 0 & -0.6154 & 0.7882 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \\ \mu_i &= \begin{bmatrix} 0 & 1.2308 & 0 \\ -1 & -1 & 0 \\ 0 & 1.5764 & 0 \end{bmatrix} \\ \mu_2 &= v \times \phi_2 \\ &= \begin{bmatrix} 0 & 0.7882 & 0.6154 \\ -1 & 0 & 0 \\ 0 & -0.6154 & 0.7882 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \\ \mu_2 &= \begin{bmatrix} 0.7882 & 1.2308 & 0.6154 \\ 0 & -2 & 0 \\ -0.6154 & 1.5764 & 0.7882 \end{bmatrix} \end{aligned}$$

3.3.7 Identifikasi

Untuk mengetahui kecocokan wajah dengan wajah yang ada didalam *database* maka diperlukan proses identifikasi. Demikian juga berlaku untuk mencocokkan waja yang baru masuk (*test face*). Nilai *eigenface* di dapat dari:

$$\begin{aligned} \mu_{new} &= v \times (\Gamma_{new} - \Psi) \\ \Omega &= [\mu_1, \mu_2, \dots, \mu_M] \end{aligned}$$

Langkah awal yang harus dilakukan adalah mencari selisih (Φ) *test face* dan nilai tengah (Ψ). Matrik *test face* koordinat tersebut dapat dilihat pada gambar 6.



Gambar 6. Citra wajah baru

$$\begin{aligned} \phi_{new} &= \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 3 \\ 1 & 0 & 3 \end{bmatrix} \end{aligned}$$

Tahap berikutnya dapat dihitung nilai *eigenface* yang di dapat dari:

$$\begin{aligned} \mu_{new} &= \begin{bmatrix} 0 & 0.7882 & 0.6154 \\ -1 & 0 & 0 \\ 0 & -0.6154 & 0.7882 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 3 \\ 1 & 0 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 0.7882 & 1.2308 & 0.6154 \\ 0 & -2 & 0 \\ -0.6154 & 1.5764 & 0.7882 \end{bmatrix} \end{aligned}$$

Untuk mendapatkan selisih terkecil antara *eigenface training image* (Γ_i) *database* dengan *eigenface test face* (Γ_{new}) dapat menggunakan *euclidean distance*, lalu jumlahkan masing-masing matriks dari *euclidean distance*.

$$\begin{aligned} \epsilon_k &= \|\Omega - \Omega_k\| \\ \epsilon_k &= \|\Omega - \Omega_{new}\| \\ \epsilon_1 &= \|\Omega - \Omega_{new}\| \\ &= \begin{bmatrix} 0.7882 & 1.2308 & 0.6154 \\ 0 & -2 & 0 \\ -0.6154 & 1.5764 & 0.7882 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1.2308 & 0 \\ -1 & -1 & 0 \\ 0 & 1.5764 & 0 \end{bmatrix} \\ &= \sqrt{(0.7882)^2 + (0)^2 + (0.6154)^2 + (1)^2 + (0)^2 + (0.6154)^2 + (0)^2 + (0.7882)^2} \\ &= 1.99 \end{aligned}$$

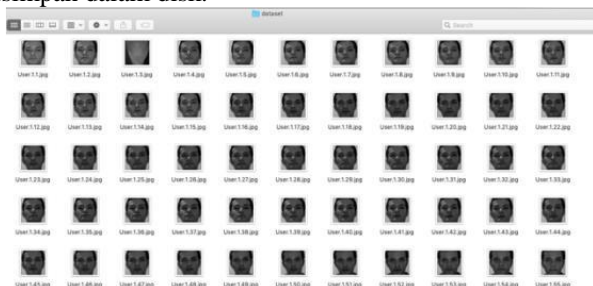
4 Implementasi dan Pengujian

Pada bagian ini, setelah perangkat lunak dibangun, dilakukan tahap uji coba. Evaluasi kinerja mencakup tingkat pengenalan, keandalan dan manajemen resource perangkat keras.

4.1 Tahap Pembuatan Database.

Pada gambar dibawah ini adalah hasil dari preprocessing untuk pembuatan database citra. Setiap citra akan diambil sampel sebanyak 100 buah dengan tujuan didapatkan database yang dihasilkan akan lebih detail. Setiap citra yang diambil ini dilakukan proses RGB ke Grayscale, konversi ke biner lalu di cropping dan menyimpannya ke dalam database. Database pada sistem ini berbentuk folder dengan kumpulan data citra hasil capture wajah.

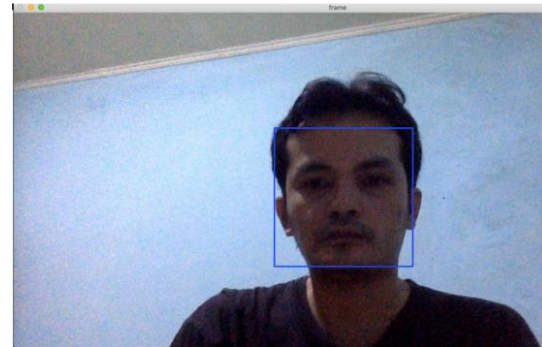
Setiap user akan di capture oleh camera pada jarak 0.35 meter sebanyak 100 gambar pada ruangan dengan intensitas cahaya terang. Berdasarkan kondisi pengujian ini, semua gambar dapat terdeteksi dan hasil capture berhasil disimpan dalam disk.



Gambar 7. Database Citra Wajah

4.2 Pengujian Deteksi Citra Wajah

Pada pengujian ini, dilakukan proses pendeteksian wajah dengan tanpa melalui proses training dahulu. Dimulai dari jarak 0.35 meter, wajah dapat dikenali dengan munculnya box berwarna biru. Dan berakhir pada jarak 0.75 meter dikarenakan pada jarak 0.75 meter box biru tidak muncul yang menandakan wajah tidak dapat dideteksi.



Gambar 8. Pengujian Deteksi Wajah

4.3 Pengujian Pengenalan Citra Wajah

Pada pengujian ini dilakukan dengan menggunakan wajah yang sama dengan database yang telah dilatih sebelumnya. Pengujian ini bertujuan untuk mengetahui apakah sistem mengenali dengan baik atau tidak citra wajah yang telah dilatih tersebut. Gambar 10 menunjukkan hasil dari pengujian ini.



Gambar 9. Hasil Pengujian Sesuai Database

Tabel di bawah ini menunjukkan hasil pengujian terhadap tiga sample citra wajah yang sesuai database dengan pengambilan citra sebanyak 100 kali tiap sample. Pengambilan data dilakukan dengan melakukan percobaan sebanyak 10 kali untuk tiap user

Tabel 1. Hasil Pengujian Pengenalan Wajah Sesuai Database

User	Percobaan Ke	Wajah Dikenali
Yunan	1	Dikenali
	2	Dikenali
	3	Dikenali
	4	Tidak Dikenali
	5	Dikenali
	6	Dikenali
	7	Dikenali
	8	Dikenali
	9	Dikenali
	10	Dikenali
Brad	1	Tidak Dikenali
	2	Tidak Dikenali
	3	Dikenali
	4	Dikenali
	5	Dikenali
	6	Dikenali
	7	Dikenali
	8	Dikenali
	9	Dikenali
	10	Dikenali
Dian	1	Dikenali
	2	Tidak Dikenali
	3	Dikenali
	4	Dikenali
	5	Dikenali
	6	Tidak Dikenali
	7	Dikenali
	8	Dikenali
	9	Dikenali
	10	Dikenali

Berdasarkan tabel diatas, dapat dihitung keakuratan sistem (As) untuk pengenalan wajah yang sesuai database dengan pendekatan rumus sebagai berikut

$$As = \frac{\text{Jumlah data benar}}{\text{Jumlah seluruh data}} \times 100\%$$

Adapun hasil keakuratan sistem untuk mengenali wajah yang sesuai database:

$$As = \frac{25}{30} \times 100\%$$

$$As = 83.33\%$$

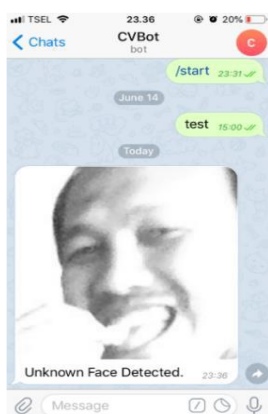
Tabel 2. Persentase keberhasilan tiap user

No	Nama User	Persentase Keberhasilan
1	Yunan	90%
2	Brad	80%
3	Dian	80%
Persentase Rata-rata		83.33%

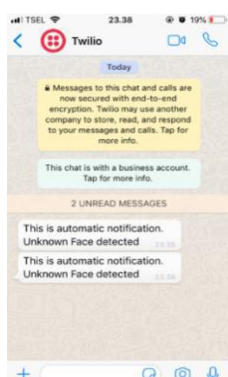
Dari hasil pengujian yang telah dilakukan didapatkan keakuratan sistem (As) sebesar 83.3% dengan 5 kali kesalahan dalam pengenalan wajah atau 16.7%. Kesalahan tersebut disebabkan oleh beberapa faktor seperti intensitas cahaya dan juga karena sistem mengenali deskriptor data sample yang paling mendekati pada database.

4.4 Pengujian notifikasi Telegram & Whatsapp

Pengujian pengenalan wajah pada bot Telegram dan Whatsapp bertujuan untuk menerima notifikasi pesan dari sistem ketika mendeteksi wajah yang tidak berada dalam database. Gambar dibawah ini menunjukkan hasil pengujian pengiriman notifikasi pada Telegram dan Whatsapp.



Gambar 10. Notifikasi ke Telegram



Gambar 11. Notifikasi ke Whatsapp

4.5 Pengujian buka pintu.

Pengujian buka pintu dilakukan oleh 3 orang dengan wajah yang tersimpan pada database dan 3 orang dengan wajah yang tidak tersimpan pada database dengan kata yang tersimpan pada database.

Pengujian dilakukan pada jarak 0.45 meter, intensitas cahaya yang bagus dan sudut 0°.

Tabel 3. Hasil Pengujian Buka Pintu

User	Training	Jarak (m)	Sudut (°)	Intensitas Cahaya (lux)	Status Pintu	Hasil
Yunan	Ya	0.45	0	350	Buka	Sesuai
Brad	Ya	0.45	0	350	Buka	Sesuai
Dian	Ya	0.45	0	350	Buka	Sesuai
Ivul	Tidak	0.45	0	350	Tutup	Sesuai
Via	Tidak	0.45	0	350	Buka	Tidak Sesuai
Ojan	Tidak	0.45	0	350	Tutup	Sesuai

Berdasarkan hasil pengujian buka pintu pada tabel 3, didapatkan bahwa sistem dapat membuka pintu dengan keakuratan sebesar 83.33% dengan hanya melakukan kesalahan satu kali.

5 Simpulan

Berdasar pada paparan diatas, dari hasil penelitian ini dapat disimpulkan hal-hal berikut ini:

- Berdasarkan pengujian sistem buka pintu, sistem dapat membuka pintu pada jarak rata-rata 0.45 meter dan rata-rata tingkat keakuratan 83.33%.
- Berdasarkan pengujian sistem buka pintu, sistem berhasil menerima user yang berada dalam database dengan akurasi mencapai 100%.
- Proses pengenalan wajah terbaik adalah pada saat citra masukan berada pada jarak 0.45 meter, sudut 0° dan intensitas cahaya 350 lux.
- Berdasarkan pengujian pengiriman notifikasi ke Telegram dan Whatsapp melalui internet, sistem berhasil mengirimkan notifikasi wajah yang tidak dikenali dengan akurasi mencapai 100%.
- Akurasi yang didapatkan sangat dipengaruhi oleh faktor intensitas cahaya, jarak, sudut pengambilan masukan citra wajah, kemiripan nilai eigenface antar user, spesifikasi kamera dan jaringan internet yang digunakan.

Kepustakaan

- BALOGH, Zoltán; MAGDIN, Martin; MOLNÁR, György. Motion Detection and Face Recognition using Raspberry Pi, as a Part of, the Internet of Things. *Acta Polytechnica Hungarica*, 2019, 16.3.
- SYED, Saifuddin, et al. Embedded System for Automatic Door Access using Face Recognition technique and Cloud Based Services. 2018.
- JAIN, Arihant Kumar; SHARMA, Richa; SHARMA, Anima. A Review of Face Recognition System Using Raspberry Pi in the Field of IoT. In: *Proceedings on International Conference on Emerg.* 2018. p. 7-14.
- SHARMA, Swati Sahu1 Prof Vijay, Internet of Things Approach for Face Detection & Face Recognition using Raspberry Pi", *International Journal for Scientific Research & Development*, 2017, Vol.5, Issue 10.
- GSPONER, David, et al. IoT: Building a Raspberry Pi security system with facial recognition. 2018.

- [6] ZEIN, Afrizal. Pendeteksian Multi Wajah dan Recognition Secara Real Time Menggunakan Metoda Principal Component Analysis (PCA) dan Eigenface. In: *ESIT*. 2018, p. 1-7.
- [7] HADIDTYO, Panji Umbar; RAHMAN, Sayuti; SUNDARI, Siti. Sistem Absensi Berbasiskan Pengenalan Wajah Secara Real Time Menggunakan Webcam dengan metode PCA (Principal Component Analysis) Eigenface dan Program Program Visual Studio. *Snastikom 2019*, 2019, 304-307.