

Aplikasi Pengenalan Wajah Untuk Membuka Pintu Berbasis *Raspberry Pi*

Axl Hanuebi, Sherwin Sompie, Feisy Kambey

Teknik Elektro Universitas Sam Ratulangi Manado, Jl. Kampus Bahu-Unsrat Manado, 95115

14021106052@student.unsrat.ac.id, aldo@unsrat.ac.id, feisykambey@unsrat.ac.id

Abstract — To prevent access to a room by people who do not have authority, a door that can be locked properly is required. Rapidly developing technology can be used to replace conventional keys that are easily damaged, broken, or duplicated. One technology that can be implemented is biometric authentication by verifying face recognition. In this research, a face recognition application was created that can open the door when the user's face is recognized. The application will be made based on raspberry pi. when the application starts, if the user's face is recognized, the servo motor that holds the door will move so that the door on the model can be opened.

Keywords — Door, Face Recognition, Local Binary Pattern, Raspberry Pi

Abstrak — Untuk mencegah akses ke dalam ruangan oleh orang yang tidak memiliki kewenangan, maka diperlukan pintu yang dapat dikunci dengan baik. Teknologi yang berkembang dapat digunakan untuk menggantikan kunci konvensional yang mudah dirusak atau diduplikasi. Salah satu teknologi yang dapat diimplementasikan adalah otentikasi biometrik dengan pengenalan wajah. Pada penelitian ini dibuatlah sebuah aplikasi pengenalan wajah yang dapat membuka pintu saat wajah pengguna dikenali. Aplikasi ini berbasis *raspberry pi*. Algoritma untuk mengenali wajah yang digunakan adalah *local binary pattern histogram*. Dalam penggunaannya, jika wajah pengguna dikenali maka *servo* yang menahan pintu bergerak agar pintu dapat dibuka. Aplikasi ini diimplementasikan pada maket pintu.

Kata kunci — Face Recognition, Local Binary Pattern, Raspberry Pi, Pintu

I. PENDAHULUAN

Untuk menjaga keamanan suatu bangunan atau ruangan agar tidak dimasuki oleh orang yang tidak memiliki kewenangan maka diperlukan kunci. Sistem untuk mengunci pintu yang konvensional masih memiliki beberapa kekurangan, misalnya mudah rusak dan dibobol serta cenderung mudah diduplikat.

Penggunaan teknologi untuk menggantikan sistem kunci pintu yang konvensional akan mempermudah serta lebih efektif. Beberapa cara yang dapat digunakan antara lain penggunaan *password*, kartu identitas, *Personal Identification Number (PIN)* dan Otentikasi Biometrik. Dibandingkan cara otentikasi yang lainnya, otentikasi dengan teknologi biometrik lebih baik karena kecil kemungkinan untuk dimanipulasi datanya. Selain itu pada cara Otentikasi lainnya, peluang untuk lupa dan juga hilang.

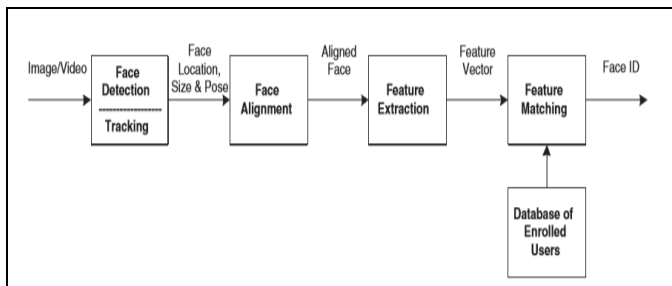
Sistem biometrik adalah sistem identifikasi pribadi dengan menentukan keaslian dari suatu karakteristik *fiologis*, ataupun perilaku tertentu yang dimiliki oleh pengguna [1]. Objek – objek untuk melakukan otentikasi dengan teknologi biometrik diantaranya adalah pemindaian retina dan iris, pemindaian sidik jari, garis telapak tangan, pemindaian wajah dan pengenalan suara. Setiap objek yang digunakan memiliki sensor yang berbeda – beda serta tingkat akurasi yang berbeda pula. Cara kerja teknologi biometrik yaitu dengan menggunakan teknik, *pattern recognition*. Objek biometrik akan menjadi dataset bagi komputer untuk membuat pola tertentu. Pola tersebut akan menjadi pembanding saat dilakukan proses otentikasi.

Salah satu objek biometrik yang dapat digunakan untuk melakukan verifikasi adalah wajah. Pengenalan wajah atau *face recognition* adalah teknologi otentikasi biometrik dengan menggunakan citra wajah sebagai pembanding. Dalam pengaplikasiannya sistem pengenalan wajah memiliki tingkat akurasi yang tidak setinggi otentikasi *fingerprnt* dan pengenalan iris mata., namun berdasarkan penelitian, rata – rata tingkat keberhasilan pengenalan wajah dengan metode *adaboost* dan *Eigenface* PCA mencapai 80% pada berbagai kondisi berbeda (jarak objek dengan sensor, pencahayaan, posisi, atribut, dan mimik muka) [2]. Sistem pengenalan wajah juga tidak membutuhkan biaya yang tinggi saat diimplementasikan karena hanya memerlukan sensor berupa kamera.

Untuk itu, berdasarkan masalah yang telah dijabarkan sebelumnya, maka dirancang dan dibangun suatu aplikasi yang dapat membuka kunci pintu secara otomatis setelah dilakukan pengenalan wajah. Sistem akan menerima input dari kamera, kemudian melakukan pendeteksian terhadap wajah. Wajah yang berhasil dideteksi kemudian akan dibandingkan dengan data yang ada. Jika wajah tersebut berhasil dikenali maka pintu yang terkunci akan dibuka secara otomatis.

A. Face Detection

Face detection adalah tahap awal pada suatu sistem pengenalan wajah. deteksi wajah mempunyai pengaruh besar pada performa dan daya guna suatu sistem pengenalan wajah. Ketika sistem pengenalan wajah menerima masukan berupa gambar atau *video*, idealnya sistem akan mengidentifikasi posisi wajah, ukuran, dan orientasi wajah yang terdapat pada gambar atau *video* tersebut. Kecepatan dan akurasi pada sistem *face detection* sangat bergantung pada intensitas cahaya dan kualitas *input* yang diterima.



Gambar 1. Proses face recognition

Parameter untuk mendeteksi wajah pada umumnya berdasarkan warna kulit, pergerakan wajah (untuk *video*), ukuran dan bentuk wajah/kepala, atau kombinasi dari parameter – parameter tersebut. Setiap citra yang diterima akan di *scan* untuk diklasifikasikan sebagai wajah atau bukan wajah. Klasifikasi sebagai wajah atau bukan wajah pada citra gambar input dapat dilakukan karena sistem sebelumnya telah di *training* menggunakan contoh – contoh data wajah dan bukan wajah. *Training* ini dilakukan menggunakan metode pembelajaran statistik [3].

B. Face Recognition

Wajah dapat digunakan sebagai salah satu objek biometrik pada tubuh manusia yang dapat digunakan untuk melakukan otentikasi. Proses otentikasi biometrik dengan menggunakan wajah sebagai pembandingan ini lebih dikenal dengan istilah *Face Recognition*.

Teknologi pengenalan wajah mengalami peningkatan secara signifikan sejak pertama kali dikembangkan. Selain itu, deteksi wajah, ekstraksi *fitur*, dan pengenalan wajah sekarang tidak hanya dilakukan pada gambar atau *video* biasa, namun dapat juga dilakukan secara langsung pada citra *video real time* yang diambil dalam situasi yang tepat.

Secara umum sistem pengenalan wajah memiliki 4 tahap utama, yaitu *detection*, *alignment*, *feature extraction*, dan *matching* [3]. *Detection* dan *alignment* merupakan dua proses awal yang harus dilakukan terlebih dahulu sebelum pengenalan wajah (*feature extraction* dan *matching*) dilakukan. Proses pada 4 tahap ini dapat dilihat lebih jelas pada gambar 1.

Pada tahap *detection* wajah akan dipisahkan dari latar belakang dan bagian gambar lain yang bukan wajah. Pada tahap *alignment* diperlukan agar lokasi dan skala setiap wajah yang terdeteksi lebih akurat. *Fitur –fitur* pada wajah seperti mata, hidung, mulut, dan garis wajah akan diukur secara *geometry* ukuran, lokasi dan posenya. Wajah juga akan dinormalisasikan berdasarkan pencahayaan dan skala keabu – abuan.

Setelah wajah dinormalisasikan pada tahap sebelumnya untuk memperoleh informasi yang efektif dan berguna untuk membedakan wajah orang yang berbeda, selanjutnya hasil ekstraksi *fitur* dari wajah yang dimasukan akan dibandingkan dengan wajah – wajah yang ada pada database. Jika ditemukan kecocokan maka wajah tersebut dikenali, jika sebaliknya maka wajah tersebut tidak dikenali. Hasil pengenalan wajah sangat bergantung pada *fitur* yang diekstraksi untuk mewakili

pola wajah dan metode klasifikasi yang digunakan untuk membedakan antara wajah satu dengan wajah yang lain. Sedangkan lokalisasi wajah dan normalisasi adalah dasar untuk mengekstraksi *fitur* yang efektif

C. OpenCV

OpenCV (*Open Source Computer Vision Library*) merupakan *library* perangkat lunak *open source computer vision* dan *machine learning* [4]. *Open cv* memiliki lebih dari 2500 algoritma yang disediakan untuk mempercepat persepsi mesin dalam menangani *computer vision* dan *machine learning*. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak gerakan kamera, melacak objek bergerak, mengekstrak model objek 3D, menghasilkan awan titik 3D dari kamera *stereo*, mengumpulkan gambar bersama untuk menghasilkan resolusi tinggi gambar seluruh pemandangan, temukan gambar serupa dari *database* gambar, dan lain – lain. Saat ini terdapat 3 algoritma pengenalan wajah pada *open cv*, yaitu *Eigenface*, *Fisherface*, dan *Local Binary Pattern Histogram* (LBPH).

OpenCV dapat diimplementasikan dengan bahasa pemrograman C++, *python*, *java*, dan *matlab* [5]. Beberapa sistem operasi yang mendukung *opencv* antara lain, *windows*, *linux*, *android* dan *mac os*.

D. Local Binary Pattern Histogram

Local Binary Pattern (LBP) diajukan sebagai salah satu metode klasifikasi *computer vision* pada tahun 1990 oleh Li Wang [6]. Kombinasi LBP dengan *gradien berorientasi histogram* pertama kali di perkenalkan pada tahun 2009. Kombinasi ini meningkatkan performa pada dataset tertentu [7].

Local Binary Pattern (LBP) adalah operator tekstur yang sederhana namun sangat efisien yang melebeli piksel pada suatu gambar dengan cara menetapkan ambang batas pada lingkungan setiap piksel dan menganggap hasilnya sebagai angka biner [8].

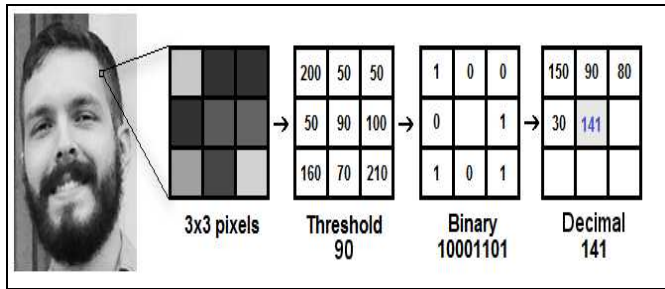
Ada beberapa tahap pada algoritma *local binary pattern histogram* [9], yaitu:

1) Parameter

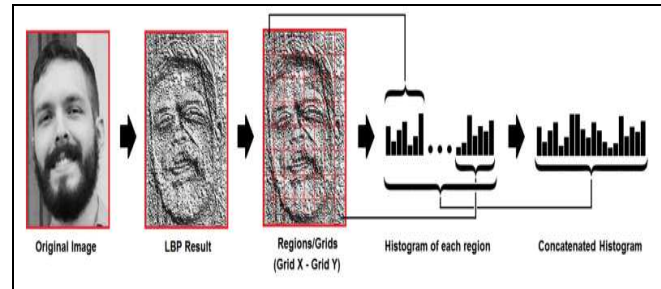
Algoritma *local binary pattern histogram* memiliki 4 parameter, yaitu *radius*, *neighbors*, *grid x*, dan *grid y*. *Radius* digunakan untuk membentuk *local binary pattern* dan merepresentasikan jari – jari di sekitar piksel tengah. Nilai dari parameter ini biasanya 1. *Neighbors* adalah jumlah titik sampel untuk membentuk *local binary pattern*. *Neighbors* biasanya bernilai 8. *Grid x* adalah jumlah sel pada arah horisontal dan *grid y* adalah jumlah sel pada arah vertikal.

2) Training algoritma

Algoritma perlu dilatih agar proses pengenalan wajah dapat dilakukan. proses ini dilakukan dengan menggunakan gambar wajah orang yang ingin dikenali sebagai dataset. Perlu juga untuk menetapkan id. Id ini bisa berupa nama atau nomor tertentu yang unik dan berbeda dengan id yang lain.



Gambar 2. Operasi *local binary pattern*



Gambar 3. Mengekstrak *histogram*

3) Menerapkan operasi *local binary pattern*.

Langkah komputasi pertama dari *local binary pattern histogram* adalah dengan membuat gambar menengah yang akan menggambar gambar asli dengan cara yang lebih baik dalam menyoroti karakteristik wajah. Proses ini dilakukan dengan menggunakan konsep *sliding window*, berdasarkan parameter *radius* dan *neighbors*.

Pada gambar 2, dapat dilihat bahwa sebuah gambar wajah sebelum melalui proses *local binary pattern* harus dibuat menjadi gambar *grayscale* terlebih dahulu. Setiap bagian pada gambar yang akan diproses, diambil dalam bentuk *window 3x3* piksel. Angka pada setiap piksel merepresentasikan intensitas cahaya bernilai 0 – 255. Kemudian nilai pada piksel yang di tengah akan menjadi ambang batas. Ambang batas ini akan menentukan 8 nilai piksel lain disekelilingnya. Jika nilai piksel di sekitarnya lebih dari nilai ambang batas, akan diubah menjadi satu, sedangkan jika kurang dari nilai ambang batas akan diubah menjadi 0. Setelah proses selesai maka semua nilai pada *window 3x3* itu adalah nilai biner. Nilai biner ini akan disusun searah jarum jam untuk dikonversikan menjadi nilai desimal. Nilai desimal akan menjadi ambang batas pada gambar baru yang akan dibentuk. Proses ini akan dilakukan secara berulang hingga semua piksel pada gambar asli dikonversi dan membentuk gambar baru yang merepresentasikan karakteristik wajah yang lebih baik dari pada gambar asli.

4) Melakukan pengenalan wajah

Gambar hasil proses sebelumnya sekarang akan dibagi menjadi beberapa *grid* dengan menggunakan parameter *grid x* dan *grid y*. Gambar 3 menunjukkan bagaimana gambar asli yang merupakan gambar *grayscale* melalui proses *local binary pattern* menjadi *histogram*. Karena gambar asli telah dibuat dalam bentuk *grayscale* maka , pada *histogram* hanya terdapat 256 posisi (0 – 255) yang mewakili intensitas kemunculan setiap piksel. Setiap *histogram dataset* akan digabungkan untuk membentuk *histogram* baru yang lebih besar, dimana *histogram* ini akan mewakili karakteristik gambar asli.

5) Melakukan pengenalan wajah

Pada tahap ini proses *trainnig* algoritma sudah selesai. Setiap *histogram* yang dibuat akan mewakili setiap gambar dari training dataset. Jadi akan diberikan gambar *input* baru yang harus melalui semua proses pada tahap sebelumnya kemudian dibandingkan kedekatannya dengan setiap *histogram* dari *dataset* hasil *training*. Dari perbandingan ini akan dikembalikan id dari *histogram* yang paling mendekati nilainya dengan *histogram* dari gambar input baru. Selain id algoritma ini juga

akan mengembalikan nilai *confidence*. Nilai *confidence* dapat digunakan untuk menentukan nilai ambang batas kedekatan *histogram*. Jika tidak membuat ambang batas maka setiap wajah yang akan dimasukan akan dikembalikan id yang nilai *histogram*nya mendekati meskipun data wajah tersebut mekipun data wajah baru tersebut belum pernah disimpan sebelumnya. Dengan menggunakan *confidece* sebagai batas maka jika gambar masukan nanti adalah wajah yang tidak dikenali, tidak akan dikembalikan nilai id yang mendekati.

E. Raspberry Pi

Raspberry pi adalah komputer berukuran kecil yang memiliki ukuran seperti kartu atm yang dapat Anda hubungkan ke tv atau layar komputer dan keyboard. Perangkat ini adalah komputer kecil yang mumpuni, dapat digunakan untuk proyek elektronik dan dapat melakukan banyak hal layaknya *PC desktop* atau komputer [10]. *raspberry pi* juga dapat digunakan sebagai pengontrol *device* pada jarak jauh maupun dekat. Alat ini dapat digunakan untuk mengotrol lebih dari satu *device*. Instruksi untuk mengontrol suatu *device* dengan *raspberry pi* dapat diimplementasikan dengan bahasa pemrograman *python*.

Raspberry Pi Foundation merupakan yayasan nirlaba yang pertama kali mengembangkan produk ini dan dirilis pada tahun 2012. Tujuan awal diproduksi *raspberry pi* adalah untuk digunakan oleh orang dewasa dan anak-anak di seluruh dunia untuk belajar pemrograman digital.

II. METODE PENELITIAN

A. Kerangka Berpikir

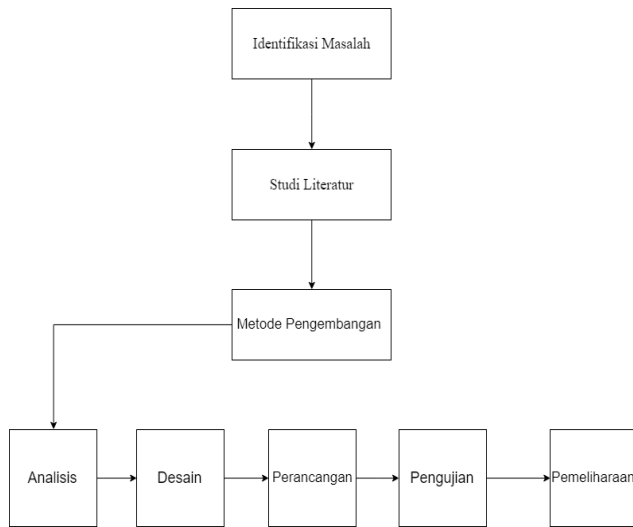
Kerangka berpikir akan menjadi acuan bagi penulis untuk melakukan penelitian agar tahapan yang dikerjakan sesuai dan tujuan dari penelitaian ini dapat tercapai. Kerangka berpikir yang menjadi acuan bagi penulis dapat dilihat pada gambar 4.

B. Perangkat Keras dan Perangkat Lunak

Perangkat keras dan perangkat lunak yang digunakan pada penelitian ini beserta spesifikasinya dan penjelasannya adalah sebagai berikut :

1) Laptop

Laptop digunakan untuk mencari informasi – informasi yang dibutuhkan pada pustaka online serta untuk merancang aplikasi yang akan dibangun. Spesifikasi laptop yang digunakan adalah sebagai berikut : sistem operasi *windows 10*, *prosesor* AMD A8 – 7410 APU, RAM 4.00 GB.



Gambar 4. Kerangka Berpikir

2) *Raspberry Pi*

Raspberry Pi digunakan sebagai perangkat utama untuk membangun aplikasi dan perangkat tempat aplikasi diimplementasikan. Spesifikasi *raspberry pi* yang digunakan adalah sebagai berikut : *raspberry pi* 3 model B, sistem operasi *raspbian ubuntu*, prosesor Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, RAM 1.00 GB.

3) *Monitor*

Monitor digunakan sebagai perangkat keras yang menampilkan antarmuka yang menghubungkan antara user dengan aplikasi.

4) *Camera Module Board REV 1.3 5MP*

Camera Module Board REV 1.3 5MP Merupakan alat untuk menerima *input frame video* agar dapat diproses oleh aplikasi. Spesifikasi *camera module board REV 1.3 5MP* adalah sebagai berikut : 5 megapixel native resolution sensor-capable of 2592 x 1944 pixel static images, mendukung 1080p30, 720p60 dan 640x480p60/90 video.

5) *Mini Servo SG-90*

Mini Servo SG-90 akan digunakan sebagai penahan pintu saat pintu tertutup dan degerakan agar pintu dapat terbuka. Spesifikasi *mini servo SG-90* adalah sebagai berikut : kecepatan tanpa loading 0.12 detik / 60 derajat (4.8V). dan *stall torque* : 1.6 kg/cm (4.8V).

6) *Python*

Python adalah bahasa pemrograman *interpretatif*. *Python* mendukung multi paradigma pemrograman, utamanya, namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman *imperatif*, dan pemrograman *fungsional*. Salah satu fitur yang tersedia pada *python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. *Python* dapat diimplementasikan pada berbagai platform dan sistem operasi. Pada penelitian ini *python* digunakan sebagai bahasa pemrograman utama untuk membangun aplikasi. Versi *python* yang digunakan pada penelitian ini adalah *python 2.7*.

7) *Fritzing*

Aplikasi *Fritzing* digunakan untuk mendesain berbagai perangkat elektronika. Pada penelitian ini *fritzing* digunakan untuk menggambar rangkaian yang diimplementasikan pada *raspberry pi*. Versi *fritzing* yang digunakan adalah versi 0.9.3b.

C. Metode Pengembangan

Metode yang akan digunakan untuk mengembangkan sistem pada penelitian ini adalah metode *waterfall*. Menurut Pressman metode *waterfall* adalah metode klasik yang bersifat sistematis, berurutan dalam membangun software [11]. Tahapan – tahapan pada model ini adalah analisis, desain, implementasi, pengujian dan pemeliharaan.

1) Analisis

Langkah pertama yang akan dilakukan oleh peneliti pada metode ini adalah dengan menganalisis dan mengumpulkan kebutuhan yang diperlukan oleh sistem yang akan dibangun. Kebutuhan sistem seperti perangkat keras dan perangkat lunak yang diperlukan sistem akan diidentifikasi pada tahap ini. Langkah ini juga diperlukan agar dapat ditentukan fitur apa saja yang perlu ada pada sistem.

2) Perancangan

Pada tahap ini peneliti akan menerjemahkan hasil dari analisis kebutuhan dan syarat pada tahap sebelumnya kedalam bentuk perancangan yang akan menjadi acuan pada tahap implementasi. Pada tahap ini akan dibuat *block diagram*, *use case*, *activity diagram*, *flowchart*, desain antar muka perangkat lunak, dan desain perangkat keras.

3) Implementasi

Pada tahap ini desain dan perancangan yang telah dibuat pada tahap sebelumnya akan diimplementasikan. Implementasi diawali dengan memasang rangkaian perangkat keras mengikuti desain perangkat keras yang telah dibuat sebelumnya. Komponen utama perangkat keras adalah *raspberry pi* yang dihubungkan dengan *monitor*, lampu *LED*, motor *servo* dan kamera. Setelah perangkat keras sudah terpasang, selanjutnya akan diinstal *library computer vision* yang bernama *opencv*. Setelah *opencv* terinstall maka akan dibuat perangkat lunak menggunakan bahasa pemrograman *python*. Aplikasi yang akan dibuat berdasarkan *block diagram*, *use case*, *activity diagram*, dan *flowchart*.

4) Pengujian

Sistem yang telah diimplementasikan pada pada tahap sebelumnya akan diuji. Pada penelitian ini aplikasi yang telah dibuat akan diuji langsung pada maket pintu seta dihubungkan dengan berbagai perangkat keras yang telah disiapkan. Pada pengujian akan kamera, motor *servo*, kemampuan perangkat lunak untuk mengenali wajah serta kecepatannya, dan jarak antara user dengan kamera saat proses pengenalan wajah dilakukan, dan intensitas cahaya yang diperlukan untuk melakukan pengenalan wajah.

5) Pemeliharaan

Pada tahap ini peneliti akan memperbaiki kesalahan – kesalahan pada sistem yang baru ditemukan pada saat sistem dijalankan dan belum pernah ditemukan pada tahap sebelumnya.

D. Use Case Diagram

Use case diperlukan akan menggambarkan bagaimana sistem aplikasi bekerja, fungsi yang ada pada sistem, serta caranya berinteraksi dengan hal – hal diluar sistem seperti pengguna atau sistem lain. Gambar 5 merupakan gambar dari use case diagram aplikasi yang dibuat pada penelitian ini.

E. Flowchart

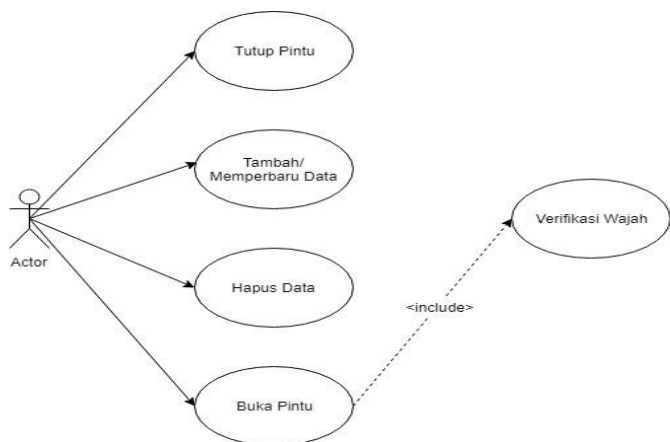
Flowchart akan menggambarkan aliran algoritma dari aplikasi yang dibuat pada penelitian ini. Ada 4 flowchart yang dibuat untuk menggambarkan aliran algoritma saat data user ditambahkan, dihapus, saat pengenalan wajah untuk membuka pintu dilakukan dan saat pintu ditutup.

Gambar 6 menunjukkan aliran proses algoritma yang terjadi pada saat data user baru ditambahkan. Proses ini dimulai dengan user memasukan id dan namanya. Setelah itu sistem akan mengecek apakah id yang dimasukan oleh user tersebut sudah ada atau belum. Jika id tersebut belum ada maka wajah user akan dicapture hingga 100 kali untuk dijadikan dataset, setelah itu id dan nama user akan disimpan pada database sqlite.

Setelah data disimpan akan dilakukan training untuk membuat model yang baru agar saat pengenalan wajah dilakukan, wajah user tersebut dapat dikenali. Jika id yang user masukan sudah ada maka, user akan diminta konfirmasi untuk memperbaru data yang user yang memiliki id tersebut. Jika user menyetujui untuk diperbaru maka wajah user akan dicapture 100 kali, id dan nama user akan disimpan pada database dan dilakukan training untuk membuat model yang baru. Jika user tidak ingin memperbarui data pada id tersebut maka proses tambah data akan terhenti.

Gambar 7 merupakan gambaran aliran algoritma yang terjadi saat hapus data dilakukan. saat user menekan tombol hapus data akan diperiksa terlebih dahulu apakah data pengguna lebih dari 0. Jika lebih dari 0 maka pengguna akan diminta untuk memasukan id user yang ingin dihapus, setelah itu akan dihapus data user pada database, dataset wajah user dan foto user sesuai id yang dimasukan. Jika data user yang ada pada database kurang dari 1 user, maka proses penghapusan data akan dibatalkan.

Pada gambar 8 dapat dilihat proses aliran algoritma yang terjadi saat user ingin menutup pintu. Proses ini diawali dengan user menekan tombol tutup pintu, setelah itu aplikasi akan mengeksekusi perintah untuk menggerakkan motor servo sehingga pintu akhirnya terkunci.



Gambar 5. Use Case

Gambar 9 berisi gambaran aliran algoritma saat user melakukan pengenalan wajah untuk membuka pintu. Pada proses ini user akan memasukan frame video real time pada aplikasi melalui kamera. Pada hasil input tersebut akan dideteksi wajah user. Apabila terdeteksi wajah maka akan dilakukan proses pengenalan wajah. Jika wajah yang terdeteksi tersebut dikenali maka akan dieksekusi perintah untuk menggerakkan motor servo agar kunci pintu terbuka

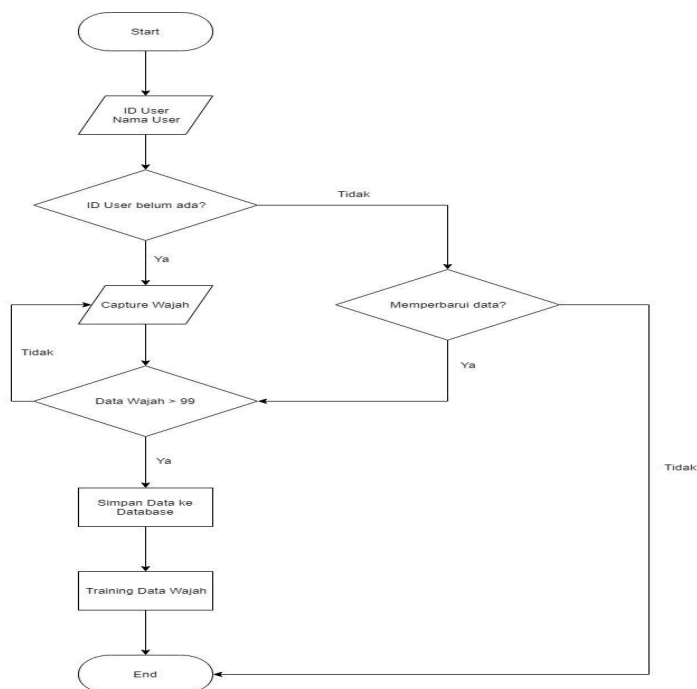
F. Perancangan Perangkat Keras

Untuk membantu peneliti agar memiliki gambaran bagaimana perangkat keras pada sistem akan diimplementasikan nantinya maka diperlukan desain skema perangkat keras. Desain ini nantinya akan menjadi acuan agar perangkat keras yang dirancang sesuai dan tidak melenceng dari tujuan awal. Desain skema perangkat keras yang dibuat pada penelitian ini menggunakan aplikasi fritzing 0.9.3b berbasis pada sistem operasi windows 64 bit. Untuk lebih jelasnya, gambar desain skema perangkat keras dapat dilihat pada gambar 10.

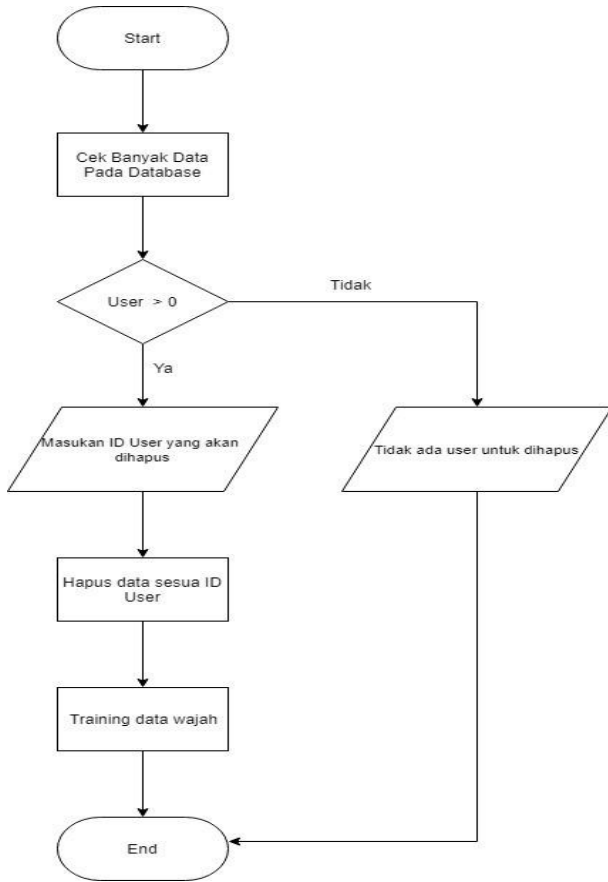
G. Block Diagram

Block diagram merupakan high level flowchart yang dibuat agar penulis dapat merancang dan menggambarkan aplikasi yang akan dibangun. Block diagram aplikasi yang dibuat pada penelitian ini dapat dilihat pada gambar 11.

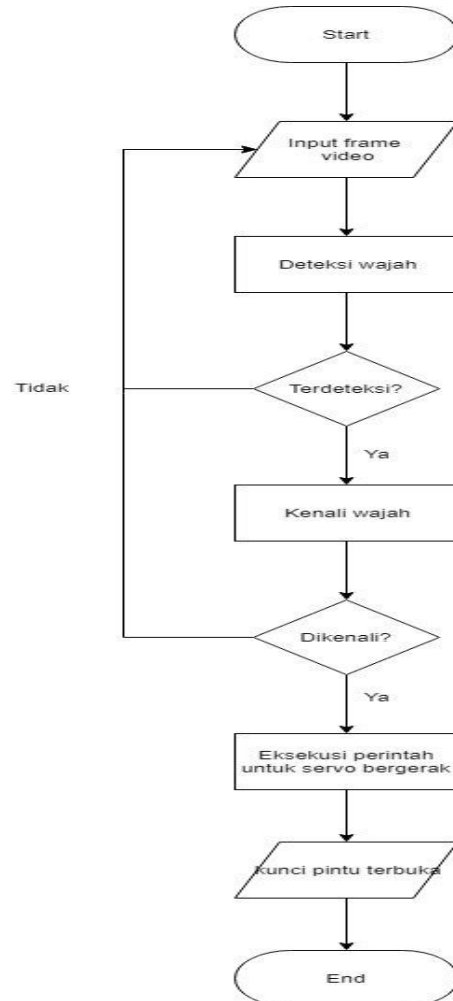
Pada gambar 11 dapat dilihat bahwa aplikasi akan menerima input berupa frame video. Input tersebut akan diproses untuk melewati tahap pre processing sebelum dilakukan deteksi wajah dengan metode haar cascade. Jika terdeteksi terdapat wajah, maka citra pada frame video akan melalui tahap pre processing untuk proses pengenalan wajah. Jika wajah pada input dikenali maka motor servo akan bergerak dan pintu akan terbuka.



Gambar 6. Flowchart Tambah Data



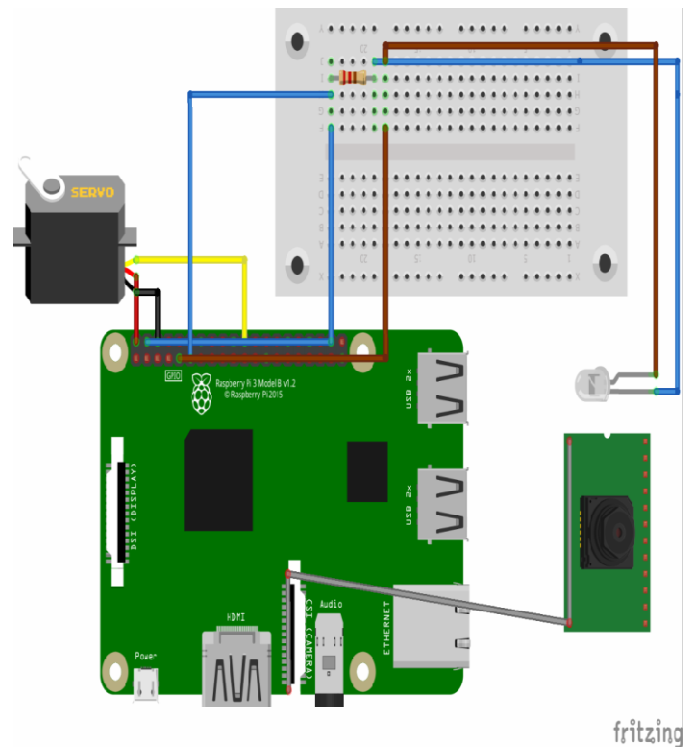
Gambar 7. Flowchart Hapus Data



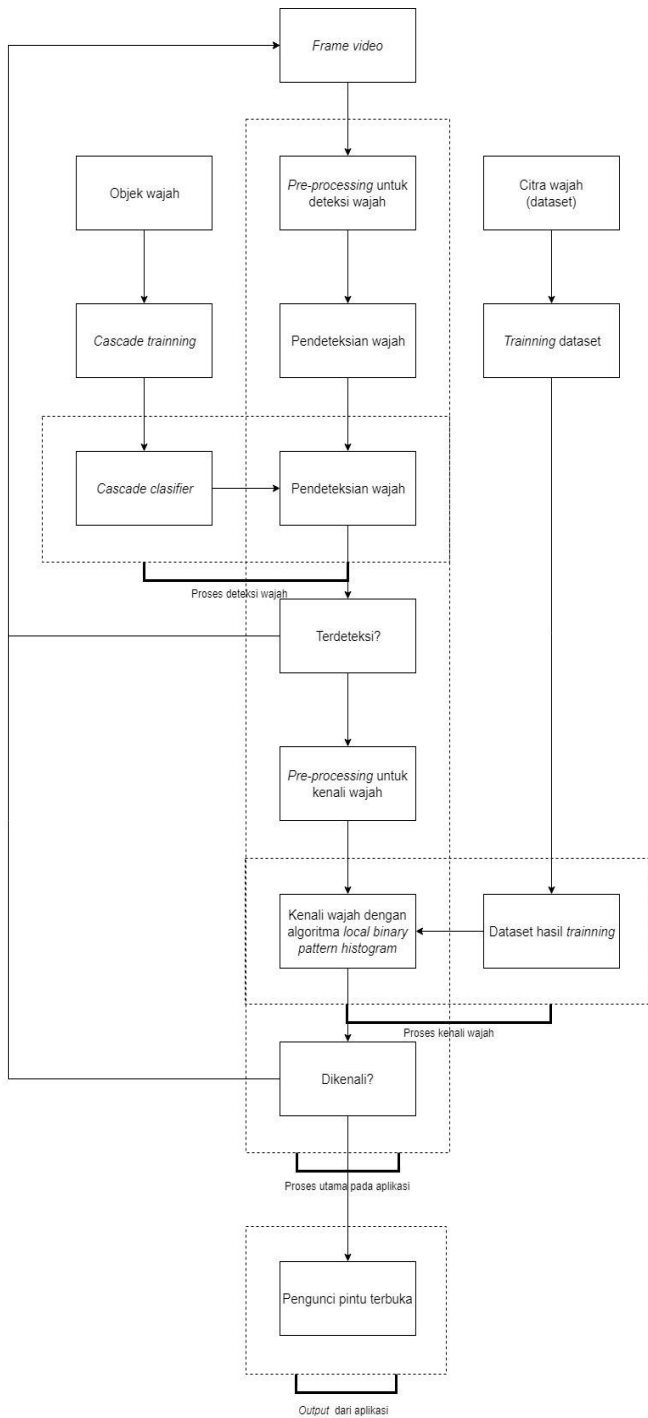
Gambar 9. Flowchart Kenali Wajah



Gambar 8. Flowchart Tutup Pintu



Gambar 10. Desain skema perangkat keras

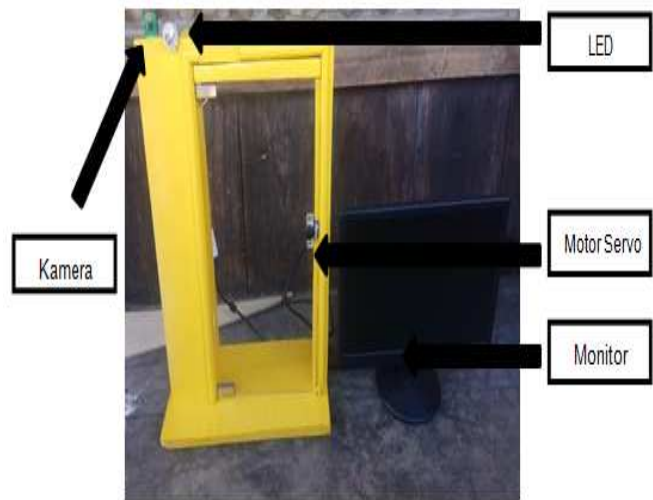


Gambar 11. Block Diagram Aplikasi

III. HASIL DAN PEMBAHASAN

A. Implementasi Perangkat Keras

Perangkat keras elektronika dan maket merupakan komponen utama pada penelitian ini karena aplikasi yang dibuat akan langsung mengendalikan dan menerima *input* dari perangkat keras yang telah terpasang pada maket berbentuk pintu. Pada tahap ini peneliti akan mengimplementasikan rancangan perangkat keras yang sudah dibuat sebelumnya menjadi sebuah rangkaian perangkat keras elektronika yang dipasang pada maket berbentuk pintu. Bentuk maket yang



Gambar 12. Tampak Depan Maket Pintu

diimplementasikan pada penelitian ini dapat dilihat pada gambar 12 dan gambar 13.

B. Implementasi Perangkat Lunak

Pada tahap ini rancangan perangkat lunak yang telah dibuat pada tahap sebelumnya akan diimplementasikan menjadi suatu perangkat lunak. Terdapat proses – proses utama yang telah diimplementasikan pada perangkat lunak yang telah dibuat. Proses – proses tersebut adalah.

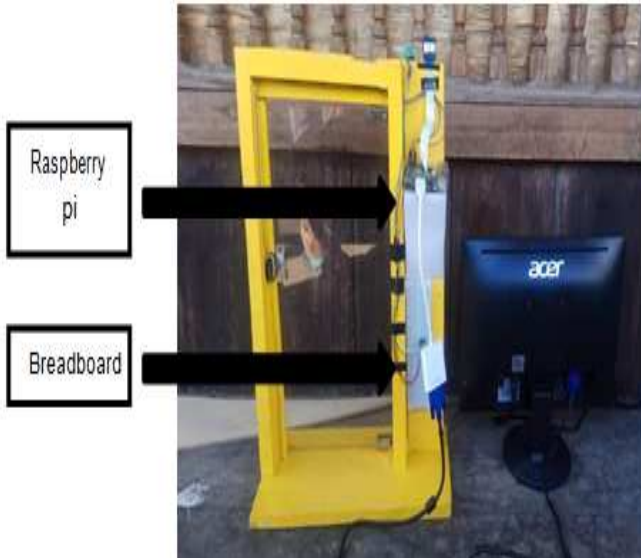
- 1) Proses tambah dan perbarui data *user*
- 2) Proses hapus data *user*
- 3) Proses *training* data
- 4) Proses pengenalan wajah

C. Antarmuka Aplikasi

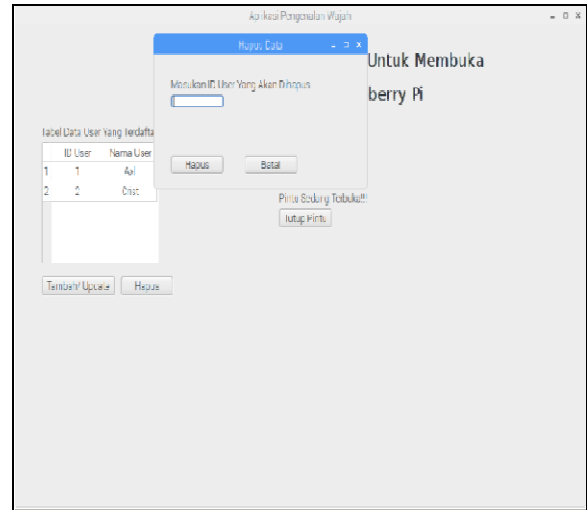
Aplikasi yang telah dibuat memiliki antarmuka *graphical user interface* sehingga saat digunakan *user* dapat langsung berinteraksi dengan aplikasi melalui *graphical user interface*. Terdapat 5 tampilan utama aplikasi yaitu, tampilan utama aplikasi saat pintu terbuka, tampilan aplikasi saat akan menambahkan atau memperbarui data *user*, tampilan aplikasi saat data *user* akan dihapus, tampilan aplikasi pintu tertutup, dan tampilan aplikasi saat wajah *user* berhasil dikenali. Setiap tampilan ini dapat dilihat pada gambar 14 samapai dengan gambar 18.

D. Pengujian

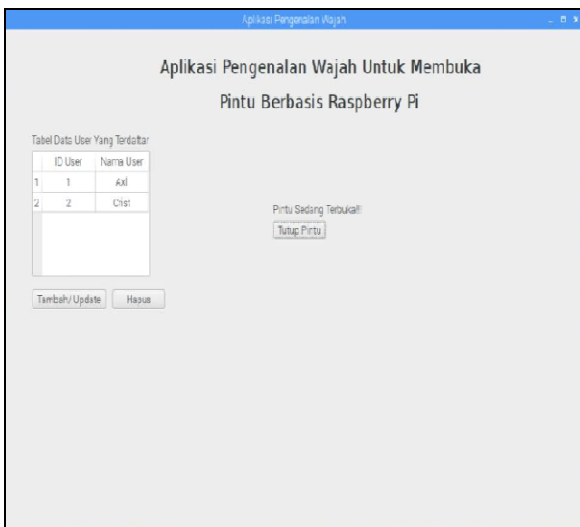
Setelah sistem didesain lalu diimplementasikan, selanjutnya sistem akan diuji apakah sudah bekerja sesuai dengan desain dan bekerja dengan baik. Pada tahap ini sistem akan melewati 6 proses pengujian yaitu, pengujian kamera, pengujian motor *servo*, pengenalan wajah, pengujian kecepatan pengenalan wajah, pengujian pencahayaan saat pengenalan wajah dan pengujian jarak saat pengenalan wajah. Hasil dari 6 pengujian yang dilakukan ini akan disusun kedalam tabel. Tabel yang menampilkan hasil pengujian yang dilakukan adalah tabel 1 sampai dengan tabel 6. Setelah pengujian, penulis akan membuat kesimpulan yang sesuai dengan hasil pengujian.



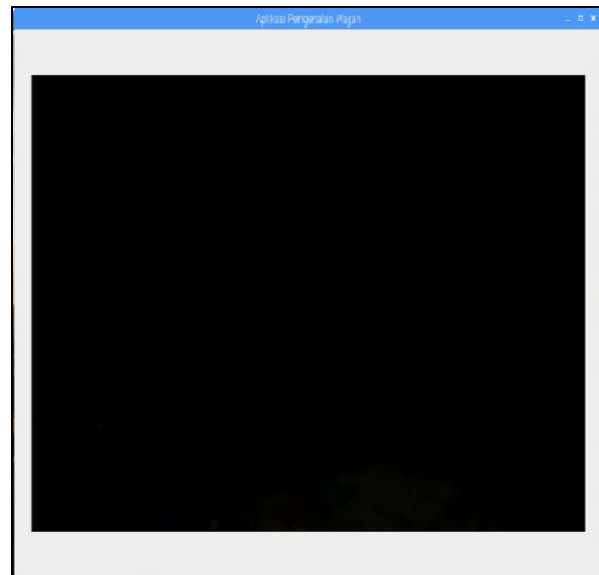
Gambar 13. Tampak Belakang Maket Pintu



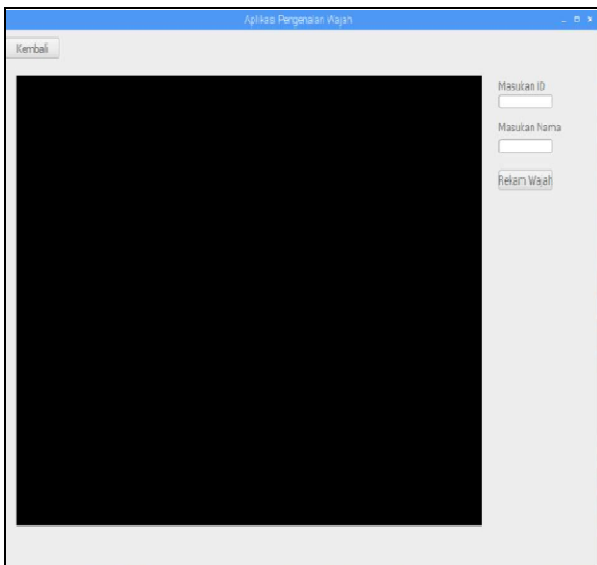
Gambar 16. Tampilan Aplikasi Saat Data *User* Akan Dihapus



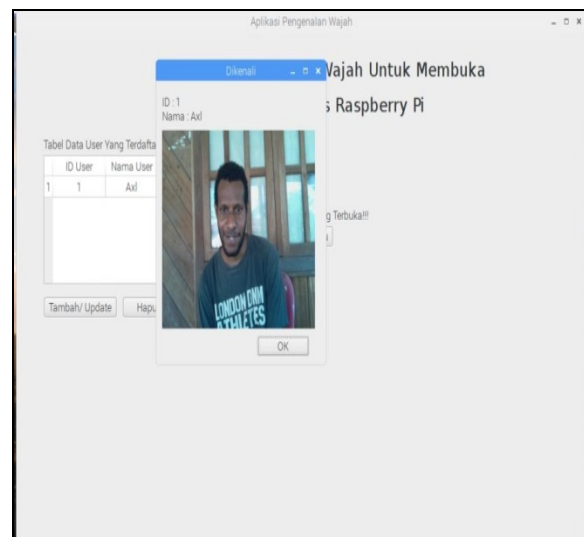
Gambar 14. Tampilan Utama Aplikasi Saat Pintu Terbuka



Gambar 17. Tampilan Aplikasi Pintu Tertutup



Gambar 15. Tampilan Aplikasi Saat Akan Menambahkan Atau Memperbaru Data



Gambar 18. Tampilan Aplikasi Saat Wajah *User* Berhasil Dikenali

TABEL I. PENGUJIAN KAMERA

No	Komponen Pengujian	Hasil yang Diharapkan	Status
1.	Kamera	Merekam dan menampilkannya hasilnya pada <i>monitor</i>	Berhasil

TABEL II. PENGUJIAN MOTOR SERVO

No	Komponen Pengujian	Hasil yang Diharapkan	Keterangan
1.	Motor Servo SG-90	Motor servo bergerak dan kembali ke posisi awal	Berhasil

TABEL III. PENGUJIAN PENGENALAN WAJAH

No	Daftar Pengujian	Skenario Uji	Keterangan
1.	Pengenalan wajah	1 user dikenali	Berhasil
2.	Pengenalan wajah	1 user dikenali 1 user tidak dikenali	Berhasil
3.	Pengenalan wajah	1 user dikenali 2 user tidak dikenali	Berhasil

TABEL IV. PENGUJIAN JARAK SAAT PENGENALAN WAJAH

No	Jarak (cm)	Dideteksi	Dikenali	Keterangan
1	20 – 39	Tidak	Tidak	Tidak dikenali
2	40 – 59	Ya	Ya	Sering dikenali
3	60 – 79	Ya	Tidak	Tidak dikenali
4	80 – 100	Ya	Tidak	Tidak dikenali

TABEL V. PENGUJIAN KECEPATAN PENGENALAN WAJAH

No	Percobaan	Waktu (second)
1	Percobaan ke-1	2.35805106163
2	Percobaan ke-2	2.33683991432
3	Percobaan ke-3	2.44685506821
4	Percobaan ke-4	2.37673306465
5	Percobaan ke-5	2.32875704765
6	Percobaan ke-6	2.36566996574
7	Percobaan ke-7	2.38476610184
8	Percobaan ke-8	2.3951280117
9	Percobaan ke-9	2.35479307175
10	Percobaan ke-10	2.29540300369
Rata –rata :		2.364299631118

TABEL VI. PENGUJIAN PENCAHAYAAN SAAT PENGENALAN WAJAH

No	Tingkat pencahayaan (Lux meter)	Dideteksi	Dikenali	Keterangan
1	< 20 Lux meter	Ya	Tidak	Jarang dideteksi
2	20 – 29 Lux meter	Ya	Tidak	Sering dideteksi
3	30 – 49 Lux meter	Ya	Ya	Jarang dikenali
4	50 – 99 Lux meter	Ya	Ya	Kadang dikenali
5	100 - 120 Lux meter	Ya	Ya	Sering dikenali

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melalui tahapan analisis, desain, implementasi dan pengujian yang telah dilakukan dapat diambil kesimpulan pada penelitian ini sebagai berikut aplikasi pengenalan wajah untuk membuka pintu berbasis *raspberry pi* berhasil dibuat.

Sistem otentikasi biometrik berupa pengenalan wajah dapat diterapkan sebagai pengganti sistem pengunci pintu konvensional.

Berdasarkan hasil pengujian perangkat keras input dan output sistem yaitu, kamera dan motor *servo* dapat bekerja dengan baik.

Berdasarkan hasil pengujian pengealan wajah, aplikasi dapat mengenali wajah user dengan kondisi user seorang diri, user dengan seseorang yang tidak dikenali, dan user dengan dua orang tidak dikenali.

Berdasarkan hasil pengujian kecepatan pengenalan wajah, rata-rata waktu yang diperlukan oleh sistem untuk dapat mengenali wajah user adalah 2.364299631118 *second*.

Berdasarkan pengujian kondisi pencahayaan maka kondisi cahaya yang paling ideal untuk melakukan pengenalan wajah pada penelitian ini adalah antara 100 – 120 lux meter.

Berdasarkan pengujian jarak maka pada penelitian ini jarak terbaik untuk melakukan pengenalan wajah adalah 40 – 60 cm.

B. Saran

Untuk penelitian selanjutnya dapat menggunakan algoritma lain untuk mendeteksi wajah dan untuk mengenali wajah.

Untuk penelitian selajutnya pengenalan wajah dapat diimplementasikan pada sistem yang lain selain untuk mengunci dan membuka pintu.

V. KUTIPAN

- [1] R. C. N. Santi, "Identifikasi Biometrik Sidik Jari dengan Metode Fraktal," *J. Teknol. Inf. Din.*, vol. 8, no. 1, pp. 68–72, 2008.
- [2] D. Suprianto, "Sistem Pengenalan Wajah Secara Real-Time," *Sist. Pengenalan Wajah Secara Real-Time dengan Adab. Eig. PCA MySQL*, 2013.
- [3] *Handbook of Face Recognition*. 2005.
- [4] A. OpenCV, "About OpenCV," *OpenCV*, 2017. .
- [5] D. R. Tobergte and S. Curtis, *OpenCV Computer Vision with Python*, vol. 53, no. 9. 2013.
- [6] Xiangsheng Huang, S. Z. Li, and Yangsheng Wang, "Shape Localization Based on Statistical Method Using Extended Local Binary Pattern," 2005.
- [7] L. Dinalankara, "Face Detection & Face Recognition Using Open Computer Vision Classifies USING OPEN COMPUTER VISION," no. August, 2017.
- [8] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006.
- [9] S. Kevin, "Face Recognition: Understanding LBPH Algorithm," 2017. [Online]. Available: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>. [Accessed: 30-Apr-2019].
- [10] Raspberry Pi Foundation, "What is a Raspberry Pi?," *raspberrypi.org*, 2010. .
- [11] R. S. Pressman, *Rekayasa Perangkat Lunak*, 7th ed. 2012.

TENTANG PENULIS



Penulis bernama Axl Hanuebi dan merupakan anak kedua dari pasangan Justinus H. Hanuebi dan Yohana Kaut, lahir di Ujung Pandang pada tanggal 25 Oktober 1995. Penulis mulai menempuh pendidikan di sekolah dasar SD YPPK St. Bonaventura Sentani (2001 – 2004) dan sekolah dasar SD St. Petrus Kanisius Gowongan Yogyakarta (2004 – 2006). Kemudian melanjutkan studi tingkat pertama di SMP YPPK St. Bonaventura Sentani (2007 – 2010) dan selanjutnya saya menempuh pendidikan tingkat atas di SMA YPPK St. Fransiskus Asisi Sentani (2010 – 2013). Setelah itu, di tahun 2014 penulis melanjutkan pendidikan ke salah satu perguruan tinggi yang berada di Manado yaitu, Universitas Sam Ratulangi Manado, dengan mengambil Program Studi S-1 Teknik Informatika di Jurusan Elektro Fakultas Teknik.