

Implementasi Face Detector sebagai Sistem Pengaman Rumah Berbasis Webcam dan Raspberry Pi

Sutarti¹, Siswanto², and Noercholith Madjid Sutardi³

^{1,2,3}Program Studi Rekayasa Sistem Komputer, Fakultas Teknologi Informasi, Universitas Serang Raya, Jl. Raya Serang – Cilegon km 5(Taman Drangong), Serang, Banten, Indonesia, 42162
e-mail: ¹sutarti86@gmail.com, ²fitraakbar06@gmail.com, ³noercholith.ms@gmail.com

Submitted Date: July 19th, 2021
Revised Date: August 12th, 2021

Reviewed Date: July 25th, 2021
Accepted Date: October 12th, 2021

Abstract

Home security is very important to protect ourselves and our wealth from the crime that is increasingly over the time. CCTV installation is an alternative method to increase home security. The disadvantages of using CCTV are not being able to send notifications to homeowners when foreigners are detected and the lack of supervision from the public and security staff. Therefore, it is necessary to develop a system that can detect the presence of unknown people and provide notifications to homeowners. In this study, implementing a face detector as a home security system using a webcam and a Raspberry Pi. The home security system that is built can recognize the faces of the occupants of the house and can open the door automatically. Face recognition uses the Haar-cascade method which compares images captured by a webcam with a database to recognize the faces of residents of the house. When the system detects the faces of the occupants of the house, the system will drive a servo motor as a door opener. The system also sends notifications via Push-Safer to homeowners when the webcam records an unrecognized face.

Keywords: Face Detector; Haar-cascade; Home Security; Notification; Raspberry Pi

Abstrak

Keamanan rumah sangatlah penting untuk melindungi diri dan barang berharga yang dimiliki dari kejahatan yang semakin marak terjadi. Pemasangan CCTV menjadi salah satu alternatif dalam meningkatkan keamanan rumah. Kelemahan dari penggunaan CCTV yaitu tidak dapat mengirim notifikasi ke pemilik rumah ketika terdeteksi orang asing dan kurangnya pengawasan dari masyarakat dan staf keamanan. Oleh karena itu perlu dikembangkan sistem yang mampu mendeteksi keberadaan orang yang tidak dikenal serta mampu memberikan notifikasi pada pemilik rumah. Pada penelitian ini mengimplementasikan *face detector* sebagai sistem pengaman rumah menggunakan Webcam dan Raspberry Pi. Sistem keamanan rumah yang dibangun dapat mengenali wajah penghuni rumah dan dapat membuka pintu secara otomatis. Pengenalan wajah menggunakan metode Haar-cascade yang membandingkan gambar yang tertangkap webcam dengan basis data untuk mengenali wajah penghuni rumah. Saat sistem mendeteksi wajah penghuni rumah maka sistem akan menggerakkan motor servo sebagai pembuka pintu. Sistem juga mengirim notifikasi melalui Push-Safer ke pemilik rumah saat webcam merekam wajah yang tidak dikenali.

Keywords: Haar-cascade; Keamanan Rumah, Notifikasi; Pengenalan Wajah; Raspberry Pi

1. Pendahuluan

Di kehidupan sekarang, keamanan sangatlah penting untuk melindungi diri dan barang berharga yang dimiliki, dari kejahatan yang semakin marak seiring berjalannya waktu. Menurut data statistik kriminal yang dikeluarkan oleh Badan Pusat Statistik pada tahun 2019, banyak kasus pencurian

yang terjadi di masyarakat dari kasus pencurian barang pribadi, pencurian kendaraan bermotor dan sebagainya tanpa penggunaan kekerasan selama tahun 2018 telah terjadi lebih dari 90.000 kasus (“Badan Pusat Statistik,” 2019). Keamanan rumah dengan memasang kamera pengawas menjadi salah satu cara untuk mengetahui segala kejadian yang

ada di rumah. Namun para pelaku tindak kriminal biasanya sudah memahami masalah teknis mengenai kamera pengawas yang berupa CCTV, sehingga oknum tersebut dapat menghilangkan data rekaman dengan cara merusak media penyimpan data rekaman, bahkan merusak kamera CCTV yang terpasang dan dapat juga membuka paksa kunci pintu rumah yang sudah terpasang. Pelaku sering kali menargetkan kejahatan di rumah yang tidak memakai sistem keamanan, menggunakan kunci pintu yang mudah dirusak dan jarang pengawasan dari masyarakat sekitar dan staf keamanan.

Dari kondisi ini maka diperlukan suatu sistem keamanan rumah dan pintu otomatis dengan mengimplementasikan *face detector* serta menggunakan *webcam* berbasis raspberry Pi. Diharapkan sistem ini dapat membantu masyarakat dalam permasalahan keamanan di rumah dan dapat melakukan pembukaan pintu otomatis saat mendeteksi wajah pemilik dan dapat juga memberikan peringatan ke pengguna melalui android sehingga mengetahui siapa saja yang berada di sekitar rumah pengguna.

Smart home dapat dibangun menggunakan sistem kendali otomatis dengan memanfaatkan modul arduino uno (Kurnianto, Hadi, & Wahyudi, 2016). Sistem yang dibuat dapat menyalakan peralatan yang ada di rumah secara otomatis, saat ada orang masuk rumah, seperti menghidupkan lampu, mengaktifkan kipas angin dan aktifitas lainnya. Sistem yang dibuat belum dapat mengenali wajah, sehingga jika ada orang lain yang dapat membuka pintu maka peralatan rumah akan tetap menyala secara otomatis.

Sistem pemantauan dan peningkatan keamanan pada pintu rumah dapat dibangun dengan mengimplementasikan pengenalan wajah dan raspberry Pi yang dihubungkan ke *cloud computing* (Natanael, Manalu, & Mulyanti, 2018). Sistem ini mampu meningkatkan keamanan pintu rumah yang menggunakan kunci konvensional dan juga tidak dapat mengirimkan pemberitahuan kepada pemilik dan penghuni rumah. Sistem cerdas ini mampu mengidentifikasi wajah pemilik rumah atau bukan, dapat mengirimkan pemberitahuan melalui *website*. Tombol yang ada pada *website* berhasil digunakan untuk membuka pintu.

Raspberry Pi yang digunakan yang terkoneksi internet dapat berfungsi sebagai kontroler yang membuat *user* dapat melakukan *monitoring* dari *smartphone* yang terkoneksi internet (Kurniawan, Kurniawan, Sunarya, & Tulloh, 2018). Sistem yang dibuat tidak dapat

mengenali manusia ataupun objek, penggunaan resolusi webcam yang kurang sehingga kualitas gambar saat monitoring berkurang.

Webcam, raspberry Pi dan library opencv dapat digunakan untuk membangun sistem keamanan dan pemantauan rumah menggunakan pengenalan wajah (Sutarti, Samsuni, & Asseghaf, 2019). Metode *Haar-cascade Classifier* yang digunakan dapat melakukan pengenalan wajah dan sistem dapat mengirimkan pesan notifikasi menggunakan *push-safer* ketika sistem mengenali wajah bukan penghuni rumah.

Bahasa pemrograman python sangat tepat digunakan pada sistem berbasis kecerdasan buatan, mesin pembelajaran (*machine learning*), *big data*, *data scientist* dan sebagainya (Enterprise, 2020). Teknologi *face detection* sudah diaplikasikan ke berbagai bidang, dan akan banyak digunakan pada teknologi masa depan (Enterprise, 2018). Penggunaan OpenCV sangat menarik karena mampu mendukung komputasi dalam aplikasi yang *real time* (Kodir, 2017).

Metode Eigenface yang merupakan sistem pengenalan wajah dapat diimplementasikan pada sistem keamanan pintu rumah (Kurniawan & Zulus, 2019). Metode Eigenface mampu mengidentifikasi orang sesuai dengan nama (label) yang ada di basis data. Eigenface tidak bisa mengidentifikasi orang yang datanya tidak tersimpan pada basis data.

Metode Fisherface merupakan metode lain yang digunakan untuk pengenalan wajah serta bisa juga diimplementasikan pada sistem keamanan pintu (Susanto, Purnomo, & Fahmi, 2017). Metode fisherface dapat mengenali wajah dan digunakan sebagai masukan pada saat akan membuka maupun mengunci pintu rumah. Namun sistem yang digunakan tidak dapat memberikan notifikasi jika ada wajah yang tidak dikenal.

Metode Triangle Face adalah metode lain yang diimplementasikan pada sistem keamanan pintu. Sistem *face recognition* ini menggunakan raspberry Pi dan dipasang pada pintu ruangan server (Wijaya, Nurhasan, & Barata, 2017). Metode *triangle face* dianggap tidak fleksibel pada saat pengambilan gambar, karena harus meletakkan dagu di penyangga dagunya. Pengambilan gambar juga dibatasi jarak.

Keamanan ruangan juga dapat memanfaatkan metode viola-jones untuk mengenali dan mendeteksi orang (Sianturi, Rahmat, & Nababan, 2018). Penempatan kamera, pencahayaan dan juga penempatan objek di suatu ruang sangat berpengaruh pada proses deteksi

objek, sehingga kadang-kadang hasilnya tidak terbaca dengan baik.

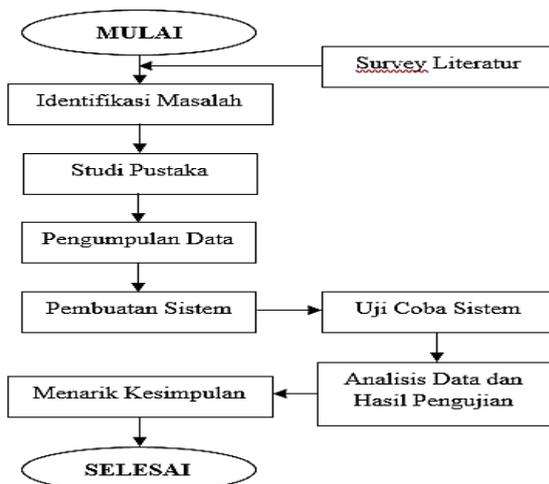
Konsep *smart room* dapat dibuat menggunakan metode *Histogram of Gradient* (HOG). Algoritma ini digunakan untuk mendeteksi manusia sebagai objeknya (Endra, Cucus, Afandi, & Syahputra, 2018). Hasil dari penelitian tersebut mampu memanfaatkan hasil deteksi keberadaan manusia sebagai masukan pada sistem *smart room*.

Ada beberapa hal yang perlu diperhatikan dalam menggunakan PCA sebagai sistem *face recognition*, yaitu banyaknya data yang digunakan sebagai data latih dan juga besaran vektor eigen (Kosasih, 2021). Penelitian ini menggunakan data sebanyak 40 orang dan masing-masing orang mempunyai 10 gambar wajah dengan posisi dan tampilan atau ekspresi yang berbeda-beda. Dari semua data gambar ini, sebagian gambar digunakan sebagai data latih, dan sisanya digunakan sebagai data uji. Tingkat akurasi sebesar 96,67% diperoleh ketika dari 10 data per orang digunakan sebanyak 7 sebagai data latih dan 3 data lainnya sebagai data uji.

2. Metodologi Penelitian

2.1 Langkah-langkah Penelitian

Langkah-langkah yang dilakukan dalam penelitian tentang sistem keamanan rumah dan pintu otomatis dengan *face detector* yang menggunakan webcam dapat dilihat pada gambar 1.



Gambar 1. Diagram Alir Penelitian

2.2 Hardware dan Software Penelitian

Penelitian ini memerlukan beberapa *hardware* antara lain:

1. Laptop
2. Raspberry Pi
3. Webcam

4. Motor Servo
5. *Smartphone*

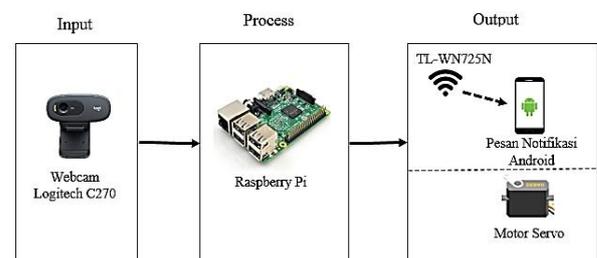
Sedangkan beberapa *software* yang diperlukan adalah:

1. PyCharm
2. Push Notifier

3. Desain dan Implementasi

Sistem keamanan rumah dan pintu otomatis dengan *face detector* menggunakan *webcam* berbasis Raspberry Pi dibangun dengan tujuan dapat mengenali wajah pemilik rumah, dapat membuka pintu secara otomatis saat mengenali wajah pemilik rumah dan juga dapat mengirimkan notifikasi ke pemilik rumah ketika ada orang yang tidak dikenali sehingga masalah-masalah mengenai sistem keamanan dan pintu otomatis dapat diselesaikan. Metode Haar cascade yang digunakan pada sistem ini merupakan salah satu dari sekian banyak algoritma *machine learning* yang banyak digunakan. Data yang berupa gambar dan video dapat dikenali wajahnya oleh metode ini sehingga haar cascade sering digunakan sebagai dasar aplikasi *object detection*.

Gambar 2 merupakan desain dari sistem keamanan rumah berbasis Raspberry Pi yang memanfaatkan *webcam* sebagai pengenalan wajah menggunakan *library* OpenCV. Masukan pada sistem menggunakan webcam Logitech C270 yang beresolusi 3 megapixel. Sedangkan keluaran pada sistem ini menggunakan TP-WN725N (Wifi USB Adapter) sebagai modul wifi. Pesan notifikasi dikirim ke *smartphone* android, dan menggerakkan Motor Servo untuk membuka pintu.



Gambar 2. Blok Diagram Sistem

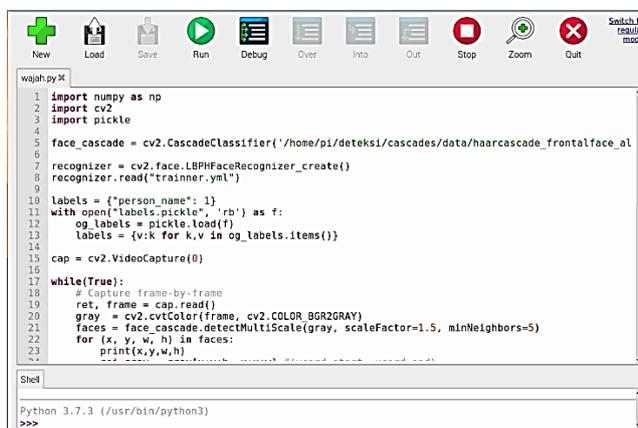
Pada diagram di atas adalah gambaran dari sistem keseluruhan di mana webcam digunakan sebagai masukan yaitu dengan merekam orang yang ada di depannya kemudian akan diproses oleh raspberry Pi jika wajah orang yang terekam dikenali maka akan menggerakkan motor servo sebagai keluaran namun jika wajah orang tersebut

tidak dikenali maka akan mengirimkan pesan notifikasi ke ponsel pintar pemilik rumah.

Sebelum sistem keamanan rumah dijalankan, terlebih dahulu merencanakan dan merancang program. Bahasa program yang akan digunakan untuk sistem pendeteksi wajah ini adalah bahasa pemrograman Python, Push-Safer API dan modul ekstra “contrib” pada OpenCV.

3.1 Python 3

Python merupakan bahasa pemrograman yang cukup banyak digunakan akhir-akhir ini. Sintaks pada python mudah dipahami dan kompatibel dengan sebagian besar sistem operasi seperti Windows, MacOS dan banyak lagi termasuk di Raspberry Pi yang berbasis Linux.



```
wajah.py
1 import numpy as np
2 import cv2
3 import pickle
4
5 face_cascade = cv2.CascadeClassifier('/home/pi/deteksi/cascades/data/haarcascade_frontalface_al
6
7 recognizer = cv2.face.LBPHFaceRecognizer_create()
8 recognizer.read("trainer.yml")
9
10 labels = {"person_name": 1}
11 with open("labels.pickle", 'rb') as f:
12     og_labels = pickle.load(f)
13     labels = {v:k for k,v in og_labels.items()}
14
15 cap = cv2.VideoCapture(0)
16
17 while(True):
18     # Capture frame-by-frame
19     ret, frame = cap.read()
20     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
21     faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)
22     for (x, y, w, h) in faces:
23         print(x,y,w,h)
```

Gambar 3. Contoh Program Python 3

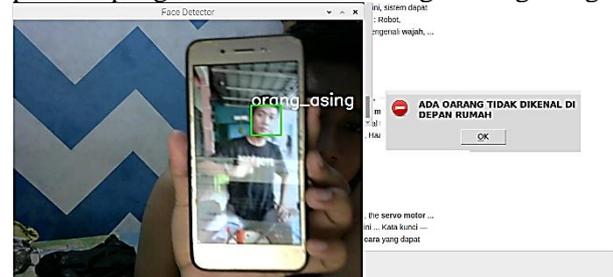
Dalam pembuatan sistem keamanan rumah ini memerlukan beberapa python package yang digunakan yaitu, NumPy, Matplotlib, Pillow, Tkinter dan Python-pushsafer.

3.2 OpenCV 4.1.0

OpenCV merupakan *library* yang diaplikasikan pada pengolahan gambar maupun video yang biasa dikenal dengan istilah *Image Processing*. Pengolahan citra yang diaplikasikan pada *Computer Vision* mampu mengenali objek, membaca teks dan angka, menghitung jumlah kendaraan yang lewat di jalan, termasuk mengenali wajah. Di dalam OpenCV ada yang sebut dengan “contrib” yang berbasis python, yang digunakan untuk mengenali wajah dan wajib menggunakan file tambahan agar sistem dapat bekerja dengan baik.

Haar-cascade pada sistem keamanan rumah ini digunakan untuk mengenali semua wajah yang terekam oleh webcam. Selain itu algoritma ini

digunakan untuk mengklasifikasikan data sebagai pemilik/penghuni rumah atau sebagai orang asing.



Gambar 4. OpenCV Menampilkan Orang Asing

3.3 Push-Safer API

Push-Safer dapat digunakan sebagai pengirim notifikasi ke banyak perangkat elektronik yang terhubung dengan koneksi internet secara *real time*, mudah dan aman. Informasi teks dikirim dalam bentuk *push notification*. Push-Safer mengirim notifikasi ke ponsel pintar pemilik rumah, di mana hasil pengolahan data yang terlebih dahulu diproses di Raspberry Pi, yang akan mengirim notifikasi jika ada orang yang tidak dikenali dengan judul notifikasi “PERINGATAN!!!” yang menginformasikan “ADA ORANG TIDAK DIKENAL DI DEPAN RUMAH”.

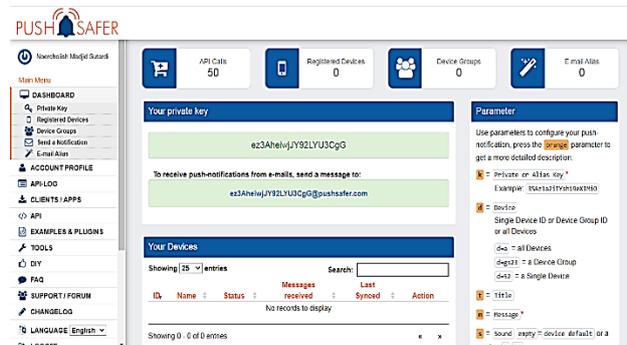
Push-Safer dapat di-*download* di situs resminya yaitu <https://www.pushsafer.com/> di mana pengguna harus terlebih dahulu mendaftarkan Akun Push-Safer yang bisa menggunakan akun gmail. Setelah itu login dengan email dan sandi yang telah didaftarkan. Informasi selanjutnya akan muncul pada tampilan Dashboard yang berupa kode unik untuk memanggil API dari Push-Safer, parameter-parameter yang digunakan, perangkat yang sudah didaftarkan, kuota yang disediakan untuk memanggil API Push-Safer dan lain sebagainya.



Sumber: https://www.pushsafer.com

Gambar 5. Cara Kerja Push-Safer

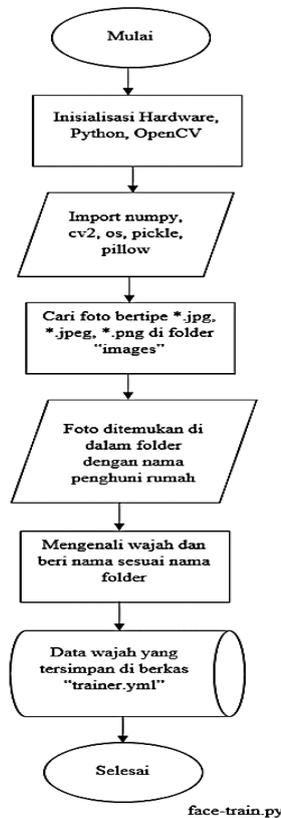
Untuk mendapatkan notifikasi ke ponsel pintar pemilik rumah harus terlebih dahulu mengunduh aplikasi Push-Safer melalui Google Play Store. Namun pengiriman notifikasi ke *smartphone* sangat dipengaruhi oleh kecepatan akses internet.



Gambar 6. Tampilan Dashboard Push-Safer

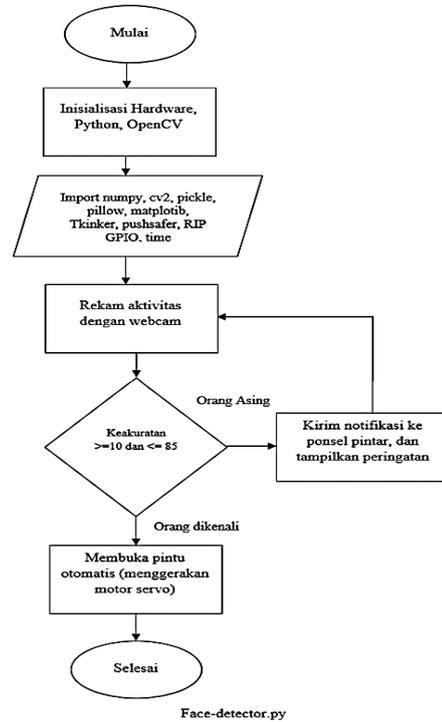
API Push-Safer atau Private Key akan digunakan ke dalam file program python untuk sistem keamanan agar dapat mengirimkan notifikasi ke perangkat yang dituju saat sistem mendeteksi orang asing di depan rumah.

3.4 Flowchart Program



Gambar 7. Flowchart Program Proses Pengenalan Wajah

Diagram alir program dari proses pengenalan wajah pada sistem terlihat pada gambar 7. Sedangkan diagram alir program dari sistem keamanan rumah terlihat pada gambar 8.



Gambar 8. Flowchart Program Sistem Keamanan Rumah

Tabel 1. Penjelasan Alur Flowchart face-train.py

Flowchart	Deskripsi
Mulai	Sistem bekerja
Inisialisasi Hardware, Python, OpenCV	Mempersiapkan perangkat keras, untuk memulai sistem termasuk perangkat lunak, Python, dan OpenCV
Import numpy, cv2, os, pickle, pillow	Penambahan package python ke dalam program yaitu pickle, numpy, pillow, cv2 dan os.
Cari foto bertipe *.jpg, *.jpeg, *.png di folder "images"	Sistem melakukan pencarian foto dengan format jpg, jpeg dan png di dalam folder untuk proses pengenalan wajah
Foto ditemukan di dalam folder dengan nama penghuni rumah	Sistem mendapatkan foto yang tersimpan di folder "penghuni rumah"
Mengenali wajah dan beri nama sesuai nama folder	Wajah yang ada di dalam folder akan dikenali dan diberi nama berdasarkan nama folder tempat foto disimpan
Data wajah yang tersimpan di berkas "trainer.yml"	Wajah yang ada di dalam folder akan dikenali dan diberi nama berdasarkan nama folder tempat foto disimpan

Tabel 2. Penjelasan Alur Flowchart face-detector.py

Flowchart	Deskripsi
Mulai	Sistem bekerja
Inisialisasi Hardware, Python, OpenCV	Mempersiapkan perangkat keras, untuk memulai sistem termasuk perangkat lunak, Python, dan OpenCV
Import numpy, cv2, pickle, pillow, matplotlib, Tkinter, pushsafer, RIP GPIO, time	Penambahan package python ke dalam program yaitu numpy, cv2, pickle, pillow, matplotlib, Tkinter, pushsafer dan menambahkan RIP GPIO yang ada di Raspberry Pi serta menambahkan time
Rekam aktivitas dengan webcam	Sistem melakukan perekaman terhadap aktivitas yang ada di depan rumah khususnya wajah manusia
Keakuratan ≥ 10 dan ≤ 85	Wajah yang terekam akan diproses oleh sistem dengan membandingkan gambar dengan data wajah yang telah disimpan terlebih dahulu, sehingga terdapat 2 kondisi yaitu: 1. Jika akurasi bernilai antara 10-85, maka sistem menganggap bahwa wajah tersebut adalah penghuni rumah 2. Jika nilai akurasi tidak sesuai maka sistem akan mengenalnya sebagai orang asing
Membuka pintu otomatis (menggerakkan motor servo)	Sistem akan menggerakkan motor servo saat sistem mengenali wajah yang tertangkap webcam
Kirim notifikasi ke ponsel pintar, dan tampilkan peringatan	Sistem akan mengirimkan push notification dan menampilkan peringatan berdasarkan hasil pengenalan wajah oleh sistem

3.5 Sistem Coding

Coding atau konstruksi program yang digunakan dalam sistem keamanan ini, dibuat dengan aplikasi *Python Interpreter* yang ada di dalam *Raspberry Pi*. Program dibuat menjadi 2 bagian utama, yang pertama program untuk memuat data wajah penghuni rumah ke dalam basis data dengan nama *faces-train.py* dan yang kedua adalah program untuk mengenali wajah secara *realtime* melalui *webcam* dengan nama *face-detector.py*.

Pada *coding* program dimuat terlebih dahulu, sehingga foto wajah penghuni rumah dibuat menjadi *database* dengan nama *trainer.yml*.

Database yang telah dibuat nantinya menjadi pembanding dalam pengenalan wajah antara gambar yang tertangkap *webcam* dengan *database*. Di bawah ini adalah foto-foto yang dijadikan *data training* dalam bentuk *jpeg, jpg, dan png*.



Gambar 9. Data training Foto Wajah “bapak”



Gambar 10. Data Training Foto Wajah “mama”



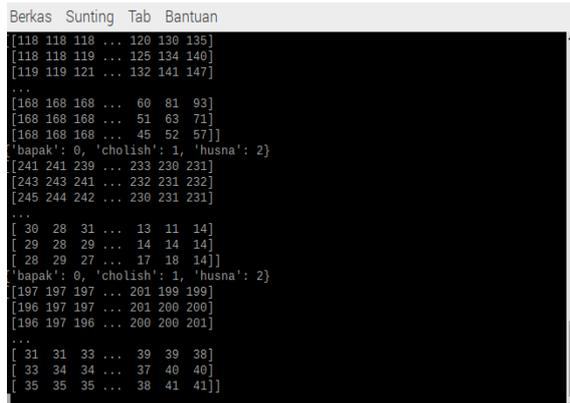
Gambar 11. Data Training Foto Wajah “husna”



Gambar 12. Data Training Foto Wajah “cholish”

Data training disimpan dalam *folder*. Nama *folder* disesuaikan dengan nama tiap penghuni rumah. Data bisa ditambahkan jika penghuni

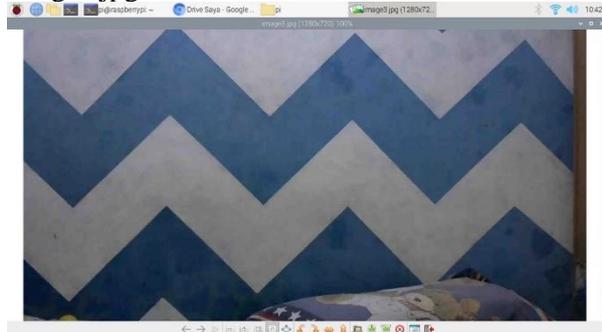
rumah bertambah. Foto wajah tersebut akan di-*training* menggunakan *faces-train.py*, untuk menjalankannya bisa melalui terminal emulator dengan perintah `sudo python3 faces-train.py` maka hasilnya akan ada angka koordinator X, Y, W, H dari setiap foto dengan tipe data `uint8` yang terlihat di terminal emulator. Angka koordinator pada setiap foto akan terbentuk data di `trainer.yml` ketika *faces-train.py* selesai dijalankan.



Gambar 13. Hasil Dari Angka Koordinat Foto

3.6 Hasil Pengujian Webcam Logitech C270

Penggunaan webcam Logitech c270 pada sistem ini bekerja dengan baik, dengan dibuktikan pada gambar 13 di mana ketika mengakses webcam pada terminal emulator di Raspberry Pi dengan perintah `fswebcam -r 1280x720 --no-banner image3.jpg`.



Gambar 14. Webcam dapat Menampilkan Hasil

3.7 Hasil Pengujian Push-Safer API & Pengiriman Notifikasi

Setelah melakukan pengujian pada *push-safer* didapat hasil pengiriman notifikasi ke ponsel pintar dapat berjalan, namun tergantung terhadap kecepatan internet saat melakukan pengiriman.

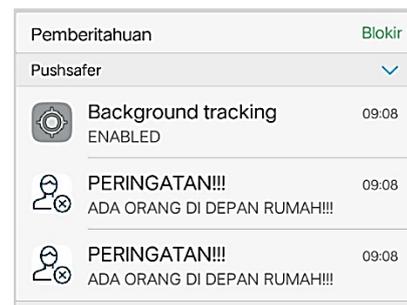


Gambar 15. Hasil Pengujian Push-Safer

Dan hasil yang didapat saat program *face-detector.py* dijalankan di terminal emulator dengan perintah `sudo python3 face-detector.py`, dapat mengirim notifikasi yang berupa teks ke ponsel pintar pemilik rumah saat webcam merekam wajah orang yang tidak dikenali dengan baik.

Tabel 3. Pengiriman Notifikasi

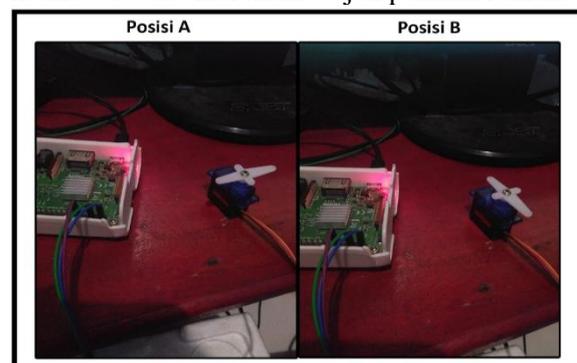
Pengujian ke	Notifikasi
1	Terkirim/sukses
2	Terkirim/sukses
3	Terkirim/sukses



Gambar 16. Notifikasi yang Diterima di Ponsel Pemilik Rumah

3.8 Hasil Pengujian PIN GPIO & Motor Servo

Hasil yang diperoleh dari pengujian *PIN GPIO* dapat bekerja dengan difungsikan sebagai *keluaran* pada sistem ini yang terhubung dengan *motor servo* dan dapat menggerakkan *motor servo*, ketika *webcam* merekam wajah pemilik rumah.



Gambar 17. Gambar Kondisi Awal (Posisi B) & Saat Mendeteksi Wajah Pemilik (Posisi A)

3.9 Hasil Pengujian Sistem

Dalam pelaksanaan pengujian sistem dilaksanakan pada dua kondisi, yaitu pada saat siang dan malam hari. Pengujian di waktu malam bertujuan untuk menguji kemampuan deteksi dengan kondisi pencahayaan yang minimum. Selain itu juga diuji dengan jarak yang divariasikan. Saat pengujian dilakukan *webcam* diletakkan di dekat pintu rumah yang mengarah ke pintu pagar rumah.

Tabel 4. Hasil Pendeteksian Webcam dari Beberapa Jarak

Jarak	Hasil Pendeteksian	Motor Servo
50 cm	Terdeteksi	Bergerak
100 cm	Terdeteksi	Bergerak
150 cm	Terdeteksi	Bergerak
200 cm	Tidak Terdeteksi	Diam

Dari hasil pengujian pendeteksian webcam dapat dilihat bahwa jarak terjauh yang dapat dijangkau oleh webcam adalah di bawah 200 cm (2m). Pengujian pada siang hari dilakukan pada pukul 13.00-14.00, sedangkan pada malam hari diuji pada pukul 21.00-22.00. Hasil pengujian dapat dilihat pada tabel 5. pada tabel selanjutnya adalah pengujian yang dilakukan berdasarkan waktu, yaitu siang hari dan malam hari.

Tabel 5. Pengujian di Waktu Siang dan Malam

Waktu	Hasil Pengujian
Siang Hari (13.00-14.00)	Berhasil/Terdeteksi
Malam Hari (21.00-22.00)	Berhasil/Terdeteksi

Pengujian selanjutnya adalah menguji keberhasilan sistem dalam mendeteksi beberapa orang sekaligus. Pengujian ini dilakukan dengan menempatkan objek dengan posisi searah dengan webcam.

Tabel 6. Pendeteksian Beberapa Orang secara Bersamaan

Banyak Orang Didepan kamera	Hasil Pengujian
1	Berhasil Dideteksi
2	Dideteksi 1 Orang
3	Dideteksi 1 Orang

Selanjutnya pengujian orang yang bergerak. Target pengujian berjalan dari pagar rumah ke arah pintu, yang terdiri dari penghuni rumah dan bukan penghuni rumah.

Tabel 7. Pendeteksian Orang berjalan ke Pintu

Pengujian ke	Objek Pengujian	Hasil Pendeteksian
1	Pemilik Rumah	Dikenali sebagai Orang Asing
2	Pemilik Rumah	Dikenali sebagai Orang Asing
3	Pemilik Rumah	Dikenali sebagai Orang Asing
4	Orang tak Dikenal	Dikenali sebagai Orang Asing

4. Pembahasan

4.1 Training Foto

Sebelum sistem dapat mengenali wajah pemilik rumah/penghuni rumah maka harus terlebih dahulu foto penghuni rumah dijadikan dalam bentuk *basis data*, di mana pada sistem ini diberi nama *trainer.yml* namun terlebih dulu harus menjalankan program *faces-train.py* terlebih dahulu.

Sebagai berkas untuk mengenali wajah yang menggunakan metode *Haar-cascade* dan *recognizer* = `cv2.face.LBPHFaceRecognizer_create()` untuk membuat basis data yang digunakan ketika pendeteksian wajah berlangsung. Dan juga kualitas dari foto yang dijadikan sebagai data *training* sangat berpengaruh terhadap hasil pendeteksian wajah.

4.2 Sistem Pendeteksian

Setelah sistem memperoleh *data training* yang dibutuhkan maka sistem bisa mengenali wajah pemilik/penghuni rumah, yang terdeteksi oleh *webcam* dengan *coding program face-detector.py* yang dapat mengenali wajah, di dalam *coding face-detector.py* ada beberapa parameter yang harus diperhatikan antara lain:

```
face_cascade = cv2.CascadeClassifier(
    '/home/pi/deteksi/cascades/data/
    haarcascade_frontalface_default.xml')
recognizer =
    cv2.face.LBPHFaceRecognizer_create()
recognizer.read("trainer.yml")
title_orangasing = "PERINGATAN !!!"
body_orangasing = "ADA ORANG TIDAK
    DIKENAL DI DEPAN RUMAH"
init ("ez3AheiwjJY92LYU3CgG")
```

Metode yang digunakan masih menggunakan *Haar-cascade* di mana dideklarasikan pada *face_cascade*. Untuk fungsi pengenalan wajah, dideklarasikan pada *recognizer*. `recognizer.read("trainer.yml")` merupakan

perintah untuk membandingkan foto yang terekam dengan basis data, kemudian `title_orangasing = "PERINGATAN!!!"` dan `body_orangasing = "ADA ORANG TIDAK DIKENAL DI DEPAN RUMAH"` adalah isi pesan yang peringatan di sistem. Untuk `init ("ez3AheiwjJY92LYU3CgG")`, merupakan perintah yang membuat sistem dapat terhubung dengan `push-safer` sehingga dapat memberikan notifikasi, `ez3AheiwjJY92LYU3CgG` merupakan `API Key` yang sebelumnya didapat ketika `login` ke web `push-safer` yang terdapat di bagian `dashboard`, namun pada setiap akun memiliki `API Key` yang berbeda-beda.

```
labels = {"person_name": 1}
with open("labels.pickle", 'rb') as f:
    og_labels = pickle.load(f)
labels = {v:k for k,v in og_labels.items() }
```

`Labels` merupakan perintah untuk memuat nama pemilik wajah yang sudah menjadi basis data dan akan dimunculkan ketika sistem bekerja mendeteksi wajah. Untuk memanggil `webcam` sebagai `masukan` menggunakan perintah `cap = cv2.VideoCapture(0)`. Saat `webcam` merekam wajah pemilik rumah / penghuni rumah atau orang asing maka akan ada perintah `if else`, di mana fungsi `if` sebagai berikut:

```
if conf >= 15 and conf <= 85:
    print(id_)
    print(labels[id_])
    font = cv2.FONT_HERSHEY_SIMPLEX
    name = labels[id_]
    color = (255, 255, 255)
    stroke = 2
    cv2.putText(frame, name, (x, y), font, 1,
color, stroke, cv2.LINE_AA)
```

dalam fungsi `if` ini terdapat kondisi `conf >= 15 and conf <= 85`: yang artinya jika persentase kecocokan wajah yang tertangkap dan wajah yang ada di basis data lebih dari sama dengan 15 dan kurang dari sama dengan 85, maka sistem akan menampilkan nama pemilik wajah penghuni rumah dengan perintah `print(id_)`. Pada fungsi `else`:

```
else:
    print('orang asing')
    font = cv2.FONT_HERSHEY_SIMPLEX
    name = 'orang asing'
    color = (255, 255, 255)
    stroke = 2
    cv2.putText(frame, name, (x,y), font, 1,
color, stroke, cv2.LINE_AA)
    messagebox.showerror(title_orangasing,
body_orangasing)
```

```
img_item="my.image.png"
cv2.imwrite(img_item, roi_color)
#isi pesan yang akan dikirim ke ponsel
Client("").send_message("ADA ORANG DI
DEPAN RUMAH!!!", "PERINGATAN!!!",
"25893", "104", "device default", "",
"https://www.pushsafer.com", "Open Pushsafer",
"", "", "", "", "", "", "", "", "")
```

Saat kondisi `if` tidak terpenuhi maka sistem akan menampilkan teks "orang asing" pada wajah yang terdeteksi, dengan perintah `print('orang asing')` dan menampilkan jendela peringatan, dan pendeteksian pun akan berhenti sampai jendela peringatan ditutup secara manual, ini dilakukan karena kuota `API Key Push-Safer` terbatas. Sistem juga akan mengirimkan notifikasi yang dilakukan oleh `Push-Safer` dengan perintah parameter `client.send_message("Message", "Title", "Device or Device Group ID", "Icon", "Sound", "Vibration", "URL", "URL Title", "Time2Live", "Image 1", "Image 2", "Image 3")`.

4.3 Analisis Motor Servo

Motor servo dapat bergerak ketika sistem mengenali wajah yang terekam `webcam`, sebelum `motor servo` dapat bekerja terlebih dahulu menambahkan beberapa parameter di file `face-detector.py`, dengan `import RPi.GPIO` kemudian menambahkan parameter,

```
servoPIN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)
pwm = GPIO.PWM(servoPIN, 50) #
GPIO 17 for PWM with 50Hz
pwm.start(0) # Initialization
```

Untuk menghubungkan raspberry pi dengan motor servo yaitu melalui pin `GPOI 17` pada pin 11 dengan perintah `servoPIN = 17` di mana motor servo digunakan sebagai keluaran `GPIO.setup(servoPIN, GPIO.OUT)` kemudian menambahkan `pwm` untuk PWM motor servo dengan frekuensi 50Hz `pwm = GPIO.PWM(servoPIN, 50)`, dan kondisi awal motor servo adalah 0 di mana tidak bergerak `pwm.start(0)`.

Perintah untuk menggerakkan motor servo melalui kondisi jika sistem menangkap wajah penghuni rumah, dengan coding program di dalam kondisi `if` seperti berikut:

```
if conf >= 15 and conf <= 85:
    print(id_)
    print(labels[id_])
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
name = labels[id_]
color = (255, 255, 255)
stroke = 2
cv2.putText(frame, name, (x, y), font, 1,
color, stroke, cv2.LINE_AA)
GPIO.keluaran(17, True)
pwm.ChangeDutyCycle(7.5)
sleep(1)
```

Jika kondisi $if\ conf \geq 15$ and $conf \leq 85$: terpenuhi maka motor servo akan bergerak dengan panjang pulsa $7.5\ pwm.ChangeDutyCycle(7.5)$ dan tidak akan kembali ke posisi semula jadi pada sistem ini hanya bisa membuka pintu secara otomatis dengan *motor servo* tidak menutup kembali secara otomatis, namun jika motor servo dikembalikan ke posisi semula dan *webcam* menangkap wajah penghuni rumah maka servo akan bergerak kembali.

4.4 Analisis Hasil

Dalam sistem keamanan, pendeteksi wajah, notifikasi yang menggunakan metode *Haar-cascade classifier* dan *push notification* untuk ponsel pintar serta *motor servo* didapat analisis data. Ada beberapa tahapan dalam sistem keamanan yaitu:

1. *Training* wajah dan pembuatan basis data.
2. Melakukan pendeteksi wajah dari *webcam* dengan *data training* dari basis data.
3. Menggerakkan *motor servo*.
4. Mengirim notifikasi ke ponsel pintar.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pengujian dan analisis dapat disimpulkan beberapa hal:

1. Sistem ini dapat dibangun menggunakan Raspberry Pi 3 Model B, Webcam dengan merk Logitech C270 yang memiliki resolusi 3 Megapixel, Motor Servo SG90, TP-Link USB Wifi Adapter. Sedangkan penggunaan *software* pada sistem ini adalah Library OpenCV, Python versi 3, metode Haar-cascade Classifier dan Push-Safer untuk mengirim informasi/notifikasi ke *smartphone*.
2. Sistem ini akan mendeteksi wajah penghuni/pemilik rumah melalui webcam kemudian menggerakkan motor servo ketika wajah dikenali, yang terhubung dengan Raspberry Pi.

3. Ketika sistem tidak mengenali wajah yang tertangkap webcam maka sistem akan mengirimkan notifikasi ke pemilik/penghuni rumah yang menggunakan notifikasi dari Push-Safer yang terhubung dengan internet. Kode API Key yang diperoleh di website Push-Safer akan digunakan sistem untuk menggunakan layanan notifikasi.

5.2 Saran

Berdasarkan pengujian dan hasil dari penelitian pada sistem keamanan rumah yang menggunakan pengenalan wajah, diperoleh saran-saran untuk meningkatkan kualitas kerja dari sistem keamanan sebagai berikut:

1. Posisi penempatan webcam diharapkan berada di ruangan yang cukup cahaya dan tidak terhalang benda sehingga pendeteksian yang dilakukan webcam dapat bekerja dengan maksimal.
2. Penggunaan API Key yang terbatas untuk layanan Notification dari Push-Safer, diharapkan menambah kuota API Calls atau mengganti layanan notifikasi tanpa batas sehingga sistem dapat bekerja tanpa khawatir kehabisan API calls.
3. Pada sistem ini penggunaan motor servo hanya untuk membuka pintu secara otomatis diharapkan ke depannya bisa menutup secara otomatis juga.

Daftar Pustaka

- Badan Pusat Statistik. (2019). Retrieved August 9, 2019, from Statistik Kriminal 2019 website: <https://www.bps.go.id/publication/2019/12/12/66c0114edb7517a33063871f/statistik-kriminal-2019.html>
- Endra, R. Y., Cucus, A., Afandi, F. N., & Syahputra, M. B. (2018). Deteksi Objek Menggunakan Histogram of Oriented Gradient (HOG) untuk Model Smart Room. *Explore: Jurnal Sistem Informasi Dan Telematika(Telekomunikasi, Multimedia Dan Informatika)*, 9(2). <https://doi.org/10.36448/JSIT.V9I2.1075>
- Enterprise, J. (2018). *Aplikasi Face Detector dan Digital Imaging dengan Python*. Jakarta: Elex media computindo.
- Enterprise, J. (2020). *Python untuk Membuat Game hingga Face Detector*. Jakarta: Elex Media Computindo.
- Kodir, A. (2017). *Dasar Raspberry PI – Panduan Praktis untuk Mempelajari Pemrograman Perangkat Keras Menggunakan Raspberry PI Model B*. Yogyakarta: CV Andi Offset.
- Kosasih, R. (2021). Pengenalan Wajah Menggunakan

- PCA dengan Memperhatikan Jumlah Data Latih dan Vektor Eigen. *Jurnal Informatika Universitas Pamulang*, 6(1), 1–6. <https://doi.org/10.32493/INFORMATIKA.V6I1.7261>
- Kurnianto, D., Hadi, A. M., & Wahyudi, E. (2016). Perancangan Sistem Kendali Otomatis pada Smart Home menggunakan Modul Arduino Uno. *JURNAL NASIONAL TEKNIK ELEKTRO*, 5(2), 260–270. <https://doi.org/10.25077/JNTE.V5N2.276.2016>
- Kurniawan, M., Kurniawan, M. I., Sunarya, U., & Tulloh, R. (2018). Internet of Things : Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 6(1), 1. <https://doi.org/10.26760/elkomika.v6i1.1>
- Kurniawan, R., & Zulus, A. (2019). Smart Home Security Menggunakan Face Recognition Dengan Metode Eigenface Berbasis Raspberry Pi. *Jurnal Sustainable: Jurnal Hasil Penelitian Dan Industri Terapan*, 8(2), 48–56. <https://doi.org/10.31629/SUSTAINABLE.V8I2.1484>
- Natanael, S., Manalu, F. R. G., & Mulyanti, S. (2018). Sistem Pengawasan dan Pengamanan Pada Pintu Rumah Menggunakan Raspberry PI Yang Terhubung Dengan Layanan Cloud Computing Serta Menggunakan Pengenalan Wajah. *Jurnal Elektro Unika Atma Jaya*, 11(1), 37–46. Retrieved from <http://ojs.atmajaya.ac.id/index.php/jte/article/view/1262/995>
- Sianturi, J., Rahmat, R. F., & Nababan, E. B. (2018). Sistem Pendeteksian Manusia untuk Keamanan Ruang menggunakan Viola – Jones. *Journal of Informatics and Telecommunication Engineering*, 1(2), 61–72. <https://doi.org/10.31289/JITE.V1I2.1424>
- Susanto, B. M., Purnomo, F. E., & Fahmi, M. F. I. (2017). Sistem Keamanan Pintu Berbasis Pengenalan Wajah Menggunakan Metode Fisherface. *Jurnal Ilmiah Inovasi*, 17(1). <https://doi.org/10.25047/jii.v17i1.464>
- Sutarti, S., Samsuni, S., & Asseghaf, I. (2019). Sistem Keamanan Rumah melalui Pengenalan Wajah Menggunakan Webcam dan Library Opencv Berbasis Raspberry Pi. *Jurnal Dinamika Informatika*, 8(2), 13–26. Retrieved from <https://jdi.upy.ac.id/index.php/jdi/article/view/37>
- Wijaya, I. D., Nurhasan, U., & Barata, M. A. (2017). Implementasi Raspberry Pi untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server dengan Pengenalan Wajah Menggunakan Metode Triangle Face. *Jurnal Informatika Polinema*, 4(1), 9–9. <https://doi.org/10.33795/JIP.V4I1.138>