

## **SISTEM PENGAWASAN DAN PENGAMANAN PADA PINTU RUMAH MENGGUNAKAN *RASPBERRY PI* YANG TERHUBUNG DENGAN LAYANAN *CLOUD COMPUTING* SERTA MENGGUNAKAN PENGENALAN WAJAH**

**Stanley Natanael<sup>1</sup>, Ferry Rippun Gideon Manalu<sup>2</sup>, Sri Mulyanti<sup>3</sup>**

<sup>1,2,3</sup>Program Studi Teknik Elektro – Fakultas Teknik

Universitas Katolik Indonesia Atma Jaya – Jakarta

e-mail: <sup>1</sup> stanleynatanael@windowslive.com, <sup>2</sup> [ferry.rippun@atmajaya.ac.id](mailto:ferry.rippun@atmajaya.ac.id),

<sup>3</sup>sri.mulyanti@atmajaya.ac.id

### **ABSTRAK**

Pada umumnya, pengamanan pintu rumah saat ini masih mengandalkan kunci konvensional yang tidak dapat memberikan peringatan kepada penghuni adanya tamu yang datang. Peringatan tersebut dapat meningkatkan kewaspadaan penghuni terhadap kondisi rumah. Berdasarkan alasan tersebut, dibutuhkan sistem yang dapat mengawasi kondisi pintu rumah dan memberikan peringatan jika ada tamu yang datang. Pada penelitian ini, sistem pengawasan dan pengamanan pada pintu rumah direalisasikan menggunakan perangkat *Raspberry Pi* sebagai mikrokomputer dan *website* sebagai halaman antarmuka penghuni dalam pengaturan dan pemantauan sistem. Pintu rumah dapat dibuka dengan 2 (dua) cara, yaitu melalui navigasi pada halaman *web* dan melalui pengenalan wajah. Hasil pengujian menunjukkan sistem dapat mengenali wajah penghuni dan bukan penghuni serta memberikan peringatan berupa notifikasi di *browser* jika ada tamu yang datang.

**Kata kunci:** *Raspberry Pi*, pengenalan wajah, *Google Firebase*, *web browser*, notifikasi

### **ABSTRACT**

*In general, security at the door now still rely on conventional keys which neither provide warning to the people of the guest. The commemoration is can increase vigilance the inhabitants of the about the condition of house. For that reason, needed a system that can keep an eye on the condition of doors of the house and give warning if there are guests who have come. In this research, their surveillance and security in the door of a house realized the use of devices *Raspberry Pi* as microcomputers and *website* as a page interface a dweller in the arrangement and monitoring system. At the door opens 2 ( two ) the way, for example through navigation on page *web* and through the facial recognition. The results of testing shows that the system can recognize the inhabitants of the and not the inhabitants of the and give warning notification in *browser* if there are guest.*

**Keywords:** *Raspberry Pi*, facial recognition, *Google Firebase*, *web browser*, notification.

## PENDAHULUAN

Keamanan pintu rumah menjadi salah satu faktor yang penting dalam lingkungan rumah saat ini. Banyak cara dalam melakukan aksi penyusup-an, contohnya pembukaan secara paksa pintu rumah. Hal ini menyebabkan semakin rawan dan mudahnya melakukan berbagai aksi penyusupan bila tak dicegah, bahkan jika telah terjadi akan sulit untuk dilacak ciri-ciri pelaku tersebut.

Dengan semakin banyaknya aksi penyusupan, maka diperlukan sistem untuk mengakomodasi kebutuhan pengawasan dan pengamanan pintu rumah bagi penghuninya. Tindakan pencegahan dapat dilakukan dengan berbagai cara, seperti menggunakan pengenalan wajah melalui *web camera* (*webcam*), menggunakan kata sandi sebagai pengganti kunci konvensional, atau memberikan notifikasi penghuni apabila ada tamu yang datang.

Adanya sistem pengawasan dan pengamanan pintu rumah, akan membantu penghuninya untuk mengamankan rumah dan memiliki rekam jejak bila terjadi aksi penyusupan sebagai barang bukti.

## TEORI PENDUKUNG

### A. *Raspberry Pi*

*Raspberry Pi* adalah salah satu *Single Board Computer* (SBC) yang populer dibandingkan beberapa SBC lainnya seperti *BeagleBone*, *Intel Galileo*, *PandaBoard*, dan *CubleBoard*. *Raspberry Pi* biasanya disingkat sebagai Raspi atau RPi. RPi pertama kali dirilis pada Februari 2012 yang dikembangkan oleh yayasan nirlaba *Raspberry Pi Foundation* [8]. *Raspberry Pi* dapat digunakan untuk mengendalikan perangkat keras yang terhubung dan menjalankan program berbasis *Python*.

### B. Bahasa Pemrograman *Python*

Bahasa pemrograman *Python* merupakan salah satu bahasa pemrograman yang digunakan untuk *Raspberry Pi*. Dalam menjalankan aplikasi, *Python* melakukan kompilasi program ke dalam bentuk pemrograman tingkat rendah (sejenis *Assembly*). Hasil kompilasi tersebut diinterpretasikan ke dalam bahasa mesin [3].

### C. *Firebase*

*Firebase* merupakan layanan *cloud computing* yang disediakan oleh perusahaan *Google* yang dapat bekerja sebagai alat untuk membangun aplikasi dinamis, *multitasking*, dan *realtime database*. *Firebase* mengembangkan sebuah *website* dengan bahasa pemrograman *Javascript*.

Beberapa fasilitas yang disediakan oleh *Firebase* untuk proyek aplikasi berbasis *website*, yaitu *authentication*, *database*, *storage*, dan *hosting* [7].

### D. *Open Source Computer Vision*

*Open Source Computer Vision* (*OpenCV*) merupakan perangkat lunak yang digunakan untuk pengolahan gambar [4]. Dalam penelitian ini, *OpenCV* berfungsi untuk menemukan gambar wajah, *cropping* gambar wajah, mengubah warna gambar menjadi abu-abu, dan menentukan besar perubahan warna *pixel* (umumnya disebut sebagai *threshold*) di gambar.

### E. *Local Binary Pattern*

*Local Binary Pattern* (LBP) merupakan metode untuk mengidentifikasi pengenalan wajah dari berbagai intensitas pencahayaan, mulai dari terang hingga redup [9][10].

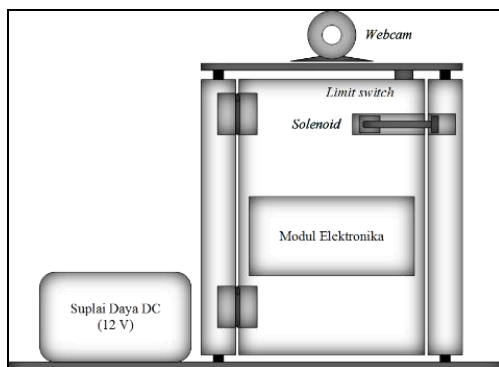
## PERANCANGAN SISTEM

Terdapat 4 (empat) bagian yang dilakukan untuk perancangan sistem,

yaitu pembangunan miniatur pintu sebagai prototipe sistem, perancangan pada *Firestore*, perancangan perangkat lunak dan *library* pada *Raspberry Pi*, dan perancangan halaman *web* sebagai halaman antarmuka untuk penghuni.

#### A. Pembangunan Prototipe

Gambar 1 menunjukkan contoh desain miniatur pintu yang dibuat sebagai prototipe dari keseluruhan sistem. Pintu yang dibangun adalah pintu berengsel yang memiliki modul elektronika, *solenoid* pada daun pintu, dan *webcam* di atas bingkai pintu tersebut. Suplai daya tambahan sebesar 12 volt digunakan untuk menyalakan *solenoid*.



Gambar 1. Ilustrasi miniatur pintu

#### B. Perancangan pada *Firestore*

Perangkat lunak *Firestore* sebagai penyedia layanan *cloud computing* memiliki 4 (empat) fasilitas yang akan digunakan pada penelitian ini, yaitu *authentication*, *database*, *storage*, dan *hosting*.

##### B.1 *Firestore Authentication*

*Firestore Authentication* ini berisikan *email address* yang bisa didaftarkan beserta dengan kata sandinya. *Email address* dan kata sandi ini digunakan untuk *login* di *website*.

##### B.2 *Firestore Database*

*Firestore Database* ini berfungsi untuk menyimpan basis data yang terhubung dengan perangkat *Raspberry Pi* dan *website*. *Firestore* dapat mengakses basis data secara *realtime* dengan syarat perangkat tersebut memiliki konektivitas internet yang stabil. Basis data memiliki 5 (lima) tabel, yaitu tabel *face*, tabel *intruder*, tabel *shutdown*, tabel *solenoid*, dan tabel *suspicious*.

Tabel *face* memiliki 5 (lima) atribut, yakni *face\_counter*, *face\_delete*, *face\_empty*, *face\_failed*, *face\_register*. Atribut *face\_counter* berfungsi untuk menampilkan banyaknya jumlah gambar wajah yang telah diambil sebagai data gambar wajah penghuni yang baru. Atribut *face\_delete* berfungsi untuk menghapus data gambar wajah. Atribut *face\_empty* berfungsi untuk menyatakan ada atau tidaknya *file trainingData.yml* di RPi. Atribut *face\_register* berfungsi untuk mendaftarkan gambar wajah ke dalam basis data. Atribut *face\_failed* berfungsi untuk menyatakan pengambilan gambar wajah gagal dilakukan.

Tabel *intruder* memiliki 2 (dua) atribut, yaitu *intruder\_image* dan *intruder\_status*. Atribut *intruder\_image* berfungsi untuk menyimpan nama *file* gambar pe-nyusup yang terakhir diunggah. Atribut *intruder\_status* berfungsi untuk mengaktifkan notifikasi di *browser* bila terjadi aksi penyusupan.

Tabel *shutdown* memiliki sebuah atribut, yaitu *shutdown\_status*. Atribut *shutdown\_status* berfungsi untuk mematikan perangkat *Raspberry Pi*.

Tabel *solenoid* memiliki sebuah atribut, yaitu *solenoid\_status*. Atribut *solenoid\_status* berfungsi untuk mengaktifkan perangkat keras *solenoid*.

Tabel *suspicious* memiliki 2 (dua) atribut, yaitu *suspicious\_image* dan *suspicious\_status*. Atribut

*suspicious\_image* berfungsi untuk menyimpan nama *file* gambar wajah tamu yang terakhir diunggah. Atribut *suspicious\_status* berfungsi untuk mengaktifkan notifikasi pada *browser* adanya tamu yang datang ke rumah.

### B.3 *Firestore Storage*

*Firestore Storage* memuat beberapa gambar yang ditangkap oleh *webcam* berkaitan aksi penyusupan. RPi mengetahui aksi penyusupan tersebut, jika perangkat keras *limit switch* dalam kondisi terbuka, sedangkan perangkat *solenoid* dalam kondisi menutup akses pintu. Sehingga, untuk melihat hasil tangkapan gambar tersebut hanya bisa dilihat pada *website Firestore* di bagian *Storage*.

### B.4 *Firestore Hosting*

*Firestore Hosting* berfungsi untuk mempublikasikan halaman *website* yang telah dibuat. Untuk melakukan *hosting* ke *cloud* digunakan *PowerShell* dan *NodeJS*.

## C. Perangkat Lunak pada *Raspberry Pi*

Untuk mempersiapkan perangkat *Raspberry Pi* agar dapat berfungsi secara optimal saat menjalankan program *Python* dibutuhkan 3 (tiga) hal, yaitu peningkatan kinerja sistem operasi *Raspbian*, pemasangan aplikasi dan *library* yang dibutuhkan, dan pembuatan program dengan menggunakan bahasa pemrograman *Python*.

### C.1 Peningkatan kinerja sistem operasi *Raspbian*

Perangkat mikrokomputer *Raspberry Pi 1 model B* memiliki memori *Central Processing Unit* (CPU) dan *Processing Unit* (GPU) yang sama. Dikarenakan pada penelitian ini lebih memanfaatkan komputasi dari CPU untuk perhitungan

algoritma LBP, pengiriman serta penerimaan data dari *Firestore*, serta pengiriman sinyal dari dan ke *pin* pada perangkat *Raspberry Pi* untuk mengendalikan perangkat yang terhubung, sehingga dibutuhkan memori CPU yang lebih besar, maka memori GPU perlu diturunkan dari 64 MB menjadi 16 MB.

### C.2 Pemasangan aplikasi dan *library* pada *Raspberry Pi*

Bahasa pemrograman yang digunakan untuk membangun perangkat lunak pada perangkat RPi adalah *Python*. Beberapa *library* yang dibutuhkan pada perangkat RPi, seperti *python-dev*, *python-rpi.gpio*, *python-pip*, *pyrebase*. Selain itu, perangkat lunak *motion* dibutuhkan untuk mengambil gambar dari *webcam* jika terjadi aksi penyusupan.

### C.3 Pembuatan program dengan bahasa *Python*

Terdapat 5 (lima) program yang difungsikan untuk mengatur kinerja sistem, yaitu *main\_program.py*, *intruder.py*, *datasetCreator.py*, *trainer.py*, dan *detector.py*. Program *main\_program.py* merupakan program utama dan program pemanggil sesuai kondisi tertentu, sedangkan program lainnya seperti *intruder.py*, *datasetCreator.py*, *trainer.py*, dan *detector.py* merupakan program yang akan dipanggil oleh program utama. Gambar 2 menunjukkan diagram alir dari program *main\_program.py* yang bertugas untuk membaca basis data di *Firestore* dan menerima sinyal masukan data dari perangkat yang terhubung ke RPi, seperti perangkat *limit switch*. Gambar 3 menunjukkan diagram alir program *intruder.py* yang bertugas untuk menangkap gambar saat terjadi aksi penyusupan. *Main\_program.py*

sebagai program utama akan bekerja dengan melakukan pengulangan terus menerus.

Untuk membuat program tersebut berjalan secara otomatis saat RPi dinyalakan, maka diperlukan *script shell* agar program `main_program.py` berjalan tanpa diperintah oleh *user*, program ini dinamakan `main_program.sh`.

#### D. Perancangan Website

*Website* yang dibuat memiliki 5 (lima) fasilitas yaitu: membuka akses pintu, mendaftarkan wajah, menghapus gambar wajah secara keseluruhan, mematikan perangkat RPi, serta mendapatkan notifikasi jika terdapat aksi penyusupan.

### PENGUJIAN SISTEM

Pengujian dilakukan dalam 3 (tiga) bagian, yakni pengujian untuk mengambil gambar wajah penghuni sebagai data, pengujian pencocokan gambar antara basis data dengan wajah penghuni untuk mendapatkan hak akses buka pintu, dan pengujian notifikasi pada *browser* jika terjadi aksi penyusupan.

#### A. Pengambilan Gambar Wajah

Gambar 4 menunjukkan contoh gambar wajah penghuni yang telah tersimpan pada basis data. Pengambilan gambar wajah dilakukan sebanyak 10 (sepuluh) kali dengan pencahayaan yang berbeda, dimulai dari terang hingga redup.

Gambar 5 menunjukkan proses pengambilan gambar di *terminal Raspberry Pi* yang dilanjutkan dengan *training data* dari sekumpulan gambar yang disimpan dengan nama `training Data.yml`.

#### B. Pencocokan Gambar

Pengujian ini dilakukan dengan membandingkan wajah penghuni atau bukan penghuni.

##### B.1 Wajah Penghuni

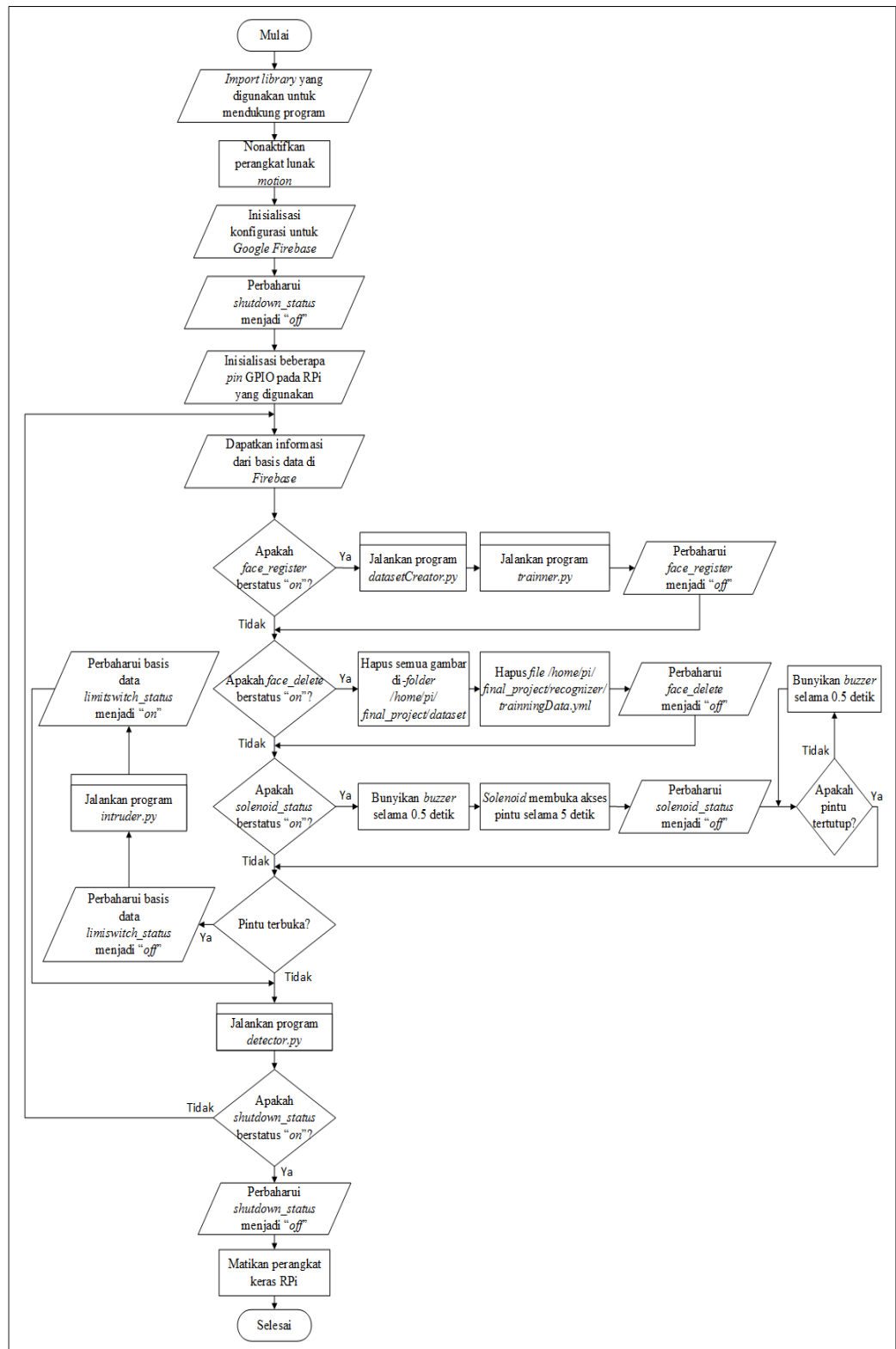
Pengujian gambar wajah penghuni pada Gambar 6 berhasil dideteksi oleh *Raspberry Pi* melalui *webcam*. Gambar 7 menunjukkan *terminal* di *Raspberry Pi* yang berhasil mengidentifikasi gambar masukan yang sesuai dengan data gambar wajah penghuni dan nilai *threshold* pada gambar masukan.

Wajah masukan yang cocok dengan data gambar wajah penghuni mengakibatkan pintu dapat dibuka.

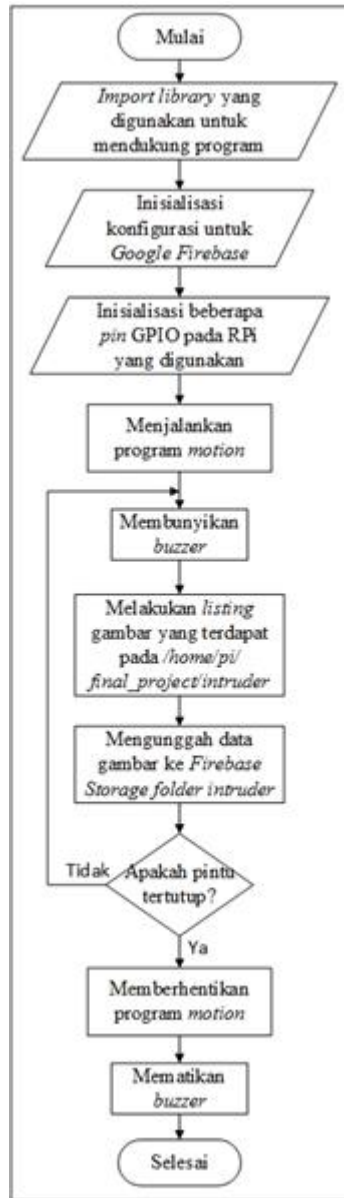
##### B.2 Wajah bukan penghuni (tamu)

Gambar 8 adalah gambar hasil identifikasi wajah bukan penghuni (tamu) di *terminal Raspberry Pi*. Setiap gambar wajah masukan yang tidak sesuai dengan basis data akan dianggap sebagai tamu. Gambar 9 merupakan contoh gambar wajah tamu yang tertangkap oleh *webcam*.

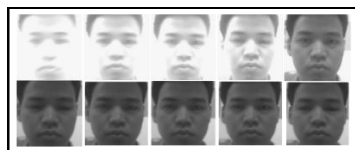
Adanya tamu yang tertangkap gambar wajahnya di *webcam*, akan memberikan notifikasi kepada penghuni melalui *browser* seperti Gambar 10 dan gambar wajahnya dapat dilihat pada *website* seperti Gambar 11.



Gambar 2. Diagram alir program main\_program.py



Gambar 3. Diagram alir program intruder.py

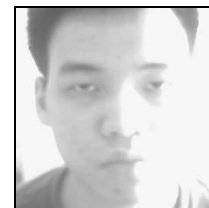


Gambar 4. Gambar wajah yang ditangkap kamera

```

pi@raspberrypi: ~/final_project
Disimpan dengan nama : dataSet/User.1.1.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.2.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.3.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.4.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.5.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.6.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.7.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.8.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.9.jpg
Tersisa 4 kali pengambilan gambar sebelum program datasetCreator.py selesai
== Face detected! ==
Disimpan dengan nama : dataSet/User.1.10.jpg
Training gambar dari path : dataSet/User.1.3.jpg
Training gambar dari path : dataSet/User.1.9.jpg
Training gambar dari path : dataSet/User.1.1.jpg
Training gambar dari path : dataSet/User.1.8.jpg
Training gambar dari path : dataSet/User.1.2.jpg
Training gambar dari path : dataSet/User.1.10.jpg
Training gambar dari path : dataSet/User.1.5.jpg
Training gambar dari path : dataSet/User.1.6.jpg
Training gambar dari path : dataSet/User.1.7.jpg
Training gambar dari path : dataSet/User.1.4.jpg
== Face Register [END] ==
    
```

Gambar 5. Proses pendaftaran gambar di terminal



Gambar 6. Wajah penghuni

```

pi@raspberrypi: ~/final_project
== Terdapat wajah yang mirip ==
Nomor ID = 1|| Nilai threshold = 41.4362317934
== Solenoid membuka akses pintu (menunggu 5 detik)==
== Solenoid menutup akses pintu ==
    
```

Gambar 7. Hasil identifikasi wajah penghuni pada terminal di Raspberry Pi

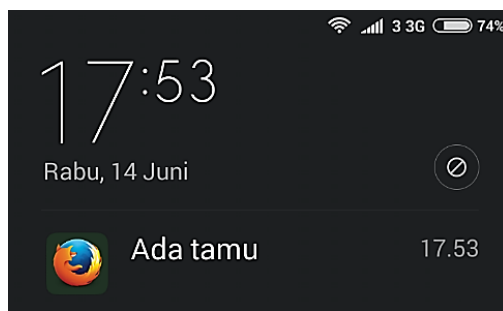
```

pi@raspberrypi: ~/final_project
==Wajah terdeteksi==
Gambar wajah masukan tidak sama dengan gambar wajah penghuni
    
```

Gambar 8. Hasil identifikasi wajah bukan penghuni



Gambar 9. Wajah bukan penghuni (tamu)



Gambar 10. Notifikasi tamu di *smartphone*

#### A. Aksi Penyusupan

Sistem akan menanggapi sebagai aksi penyusupan, jika pintu terbuka namun perangkat *solenoid* dalam keadaan tertutup.

Gambar aksi penyusupan ditangkap *webcam* dan dikirimkan ke *Firebase* seperti Gambar 12 dan sistem mengirimkan notifikasi di *browser* seperti Gambar 13.

Gambar 14 menunjukkan contoh hasil tangkapan gambar aksi penyusupan di *website*.

#### SIMPULAN DAN SARAN

Berdasarkan pengujian yang dilakukan, dapat disimpulkan bahwa:

1. Sistem ini telah berhasil digunakan untuk membuka pintu melalui tombol yang ada di *website*.



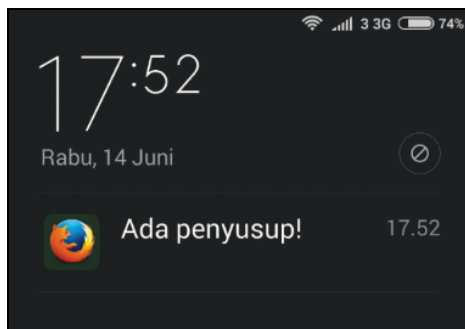
Gambar 11. Wajah tamu di *website*

2. Sistem telah berhasil untuk mendaftarkan gambar wajah penghuni ke dalam basis data.
3. Sistem ini telah berhasil mengenali wajah penghuni dan bukan penghuni.
4. Sistem ini telah berhasil memberikan notifikasi melalui *website* jika terjadi aksi penyusupan, serta mengirimkan gambar ke *Google Firebase Storage*.
5. Untuk mengoptimalkan peng-amanan pintu agar sulit dirusak oleh penyusup disarankan penggunaan perangkat keras yang lebih baik dibandingkan solenoid sebagaimana yang telah dipakai sebagai prototipe pada penelitian ini.

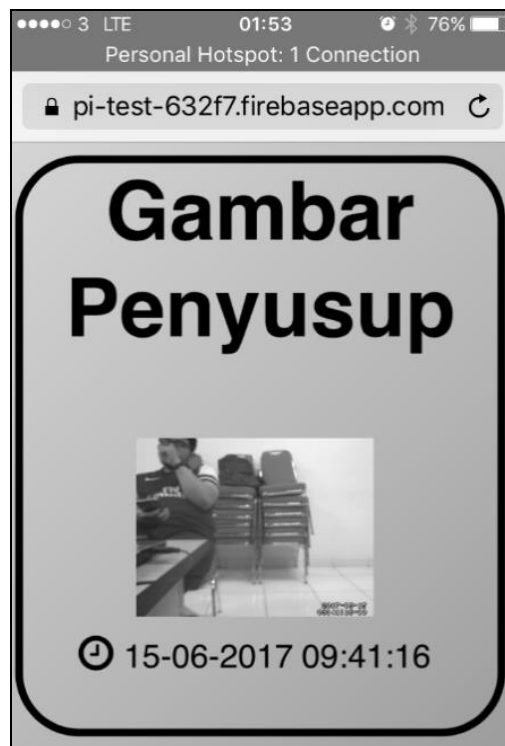


Name	Size	Type	Last modified
 01-20170522133234-01.jpg	42.42 KB	image/jpeg	22 May 2017
 01-20170522133235-00.jpg	41.13 KB	image/jpeg	22 May 2017
 01-20170522133236-01.jpg	40.85 KB	image/jpeg	22 May 2017
 01-20170522133237-00.jpg	40.79 KB	image/jpeg	22 May 2017

Gambar 12. Kumpulan gambar aksi penyusupan di *Firebase*



Gambar 13. Notifikasi adanya penyusup di *smartphone*



Gambar 14. Tampilan gambar penyusupan pada *website*

## DAFTAR PUSTAKA

- [1] Brauer, J. 2014. *Magnetic Actuators and Sensors*. United States: Wiley Publishing.
- [2] Duckett, J. 2004. *Beginning Web Programming with HTML, XHTML, and CSS*. United States: Wiley Publishing.
- [3] Hernando, K. 2014. *Perancangan dan Pembuatan Server Sistem Monitoring Keamanan Pintu Menggunakan Raspberry Pi*. Tugas Akhir. Jakarta: Fakultas Teknik Elektro, Unika Atma Jaya.
- [4] Michael, B. 2013. *OpenCV with Python Blueprints*. United Kingdom: Packt Publishing.
- [5] Richardson, M. 2016. *Make: Getting Started with Raspberry Pi*. United States: Maker Media Publisher.
- [6] Singh, B. 2003. *Control of Electrical Machines*. Mumbai: New Age International Publisher.
- [7] Stonehem, B. 2016. *Google Android Firebase: Learning the Basics*. United Kingdom: Paperback Ltd.
- [8] Tim, M. 2012. *Learning Computer Architecture with Raspberry Pi*. United States: Wiley Publisher.
- [9] Wahyudi, E., Wirawan, dan Kusuma, K. *Teknik Pengenalan Wajah Berbasis Fitur Local Binary*

*Pattern (LBP)*. Tugas Akhir.  
Surabaya: Fakultas Tek-nik Elektro,  
Universitas Institut Teknologi  
Sepuluh Nopember.

- [10] Zhao, G. 2011. *Computer Vision  
Using Local Binary Pattern*. New  
York: Springer Publisher.