

## Sistem Keamanan Rumah melalui Pengenalan Wajah Menggunakan Webcam dan Library Opencv Berbasis Raspberry Pi

**Sutarti<sup>1</sup>, Sunny Samsuni<sup>2</sup>, Isnan Asseghaf<sup>3</sup>**

<sup>1,2,3</sup> Program Studi Rekayasa Sistem Komputer, Fakultas Teknologi Informasi,  
Universitas Serang Raya, Serang - Banten  
e-mail: <sup>1</sup>sutarti86@gmail.com, <sup>2</sup>sunnysamsuni@gmail.com,  
<sup>3</sup>isnasseghaf10@gmail.com

### Intisari

Di zaman globalisasi saat ini, sistem keamanan sangatlah penting mengingat intensitas kejahatan yang semakin meningkat seiring berjalannya waktu khususnya di komplek perumahan. Masalah yang terjadi adalah pelaku kejahatan sudah mengetahui teknis dari sistem keamanan yang terpasang contohnya adalah CCTV dimana alat tersebut menggunakan DVR sebagai media penyimpanan hasil rekaman, sehingga para pelaku kejahatan dapat dengan cepat menghilangkan jejak kejahatannya dengan cara merusak media penyimpanan tersebut. Maka dari itu, penulis membangun sistem keamanan menggunakan Raspberry Pi dan *webcam* dengan pengenalan wajah. Metode yang digunakan dalam membangun alat ini adalah metode Haar-cascade Classifier sebagai metode pengenalan wajah untuk mengklasifikasikan yang mana penghuni rumah dan orang asing dan pengiriman *push notification* ke telepon pintar para penghuni rumah sebagai peringatan bahwa ada orang selain penghuni rumah berada di depan rumah.

Hasil yang didapat dari pengujian sistem keamanan yang telah dibangun menunjukkan bahwa sistem keamanan dapat mengenali baik penghuni rumah maupun orang asing dengan beberapa tes, yaitu tes pada saat siang pukul 14.18 sampai dengan 14.50 dan malam pukul 23.20 sampai 23.30, tes jarak antara *webcam* dan orang, tiga kali skenario pada saat datang kerumah baik penghuni rumah maupun orang asing, dengan beberapa orang tertangkap kamera dan notifikasi pada telepon pintar. Berdasarkan hasil pengujian dan analisis waktu siang dan malam, didapatkan persentase kesalahan untuk siang hari sebesar 35,2 %, untuk malam hari sebesar 74,3 %, waktu pendeteksian pada siang hari lebih baik dibandingkan dengan malam hari karena persentase kesalahan lebih rendah pada siang hari dibandingkan pada malam hari.

**Kata kunci**— Haar-cascade Classifier, Pengenalan Wajah, *Push Notification*, Raspberry Pi, Sistem Keamanan

## PENDAHULUAN

Di zaman yang serba canggih ini, sistem keamanan sangatlah penting mengingat intensitas kejahatan yang semakin meningkat. Pada umumnya masyarakat menggunakan sistem keamanan yang mampu merekam aktivitas di sudut rumah tertentu termasuk merekam kejadian tindak kejahatan, tanpa adanya peringatan dini. Masalah yang terjadi adalah pelaku kejahatan sudah mengetahui teknis dari sistem keamanan yang sudah terpasang yang biasanya berupa CCTV, sehingga para pelaku kejahatan dapat dengan cepat menghilangkan jejak kejahatannya dengan cara merusak media penyimpanan perekaman. Terkadang para pelaku kejahatan menargetkan rumah yang tanpa menggunakan sistem keamanan, dan rumah yang jarang dilalui masyarakat sekitar. Kurangnya pengamanan keliling atau pos ronda semakin meningkatkan aksi kejahatan yang terjadi di suatu kompleks.

Oleh karena itu, penelitian ini mengangkat tema Sistem Keamanan Rumah melalui Pengenalan Wajah Menggunakan Webcam dan Library OpenCV Berbasis Raspberry Pi. Diharapkan sistem keamanan ini dapat memberikan informasi dan peringatan dini melalui *Push Notification* bahwa terdeteksi orang yang tidak dikenal, ada di depan rumah. Sistem akan melakukan pendeteksian wajah berdasarkan wajah para penghuni rumah yang sebelumnya sudah direkam dan dimasukkan ke dalam basis data sistem, dimana yang bukan pemilik wajah akan dianggap sebagai orang tidak dikenal, lalu mengirimkan *Push Notification*. OpenCV digunakan untuk mengenali setiap wajah yang tertangkap melalui webcam.

### Kajian Pustaka

Sistem pemantauan keamanan dapat dibangun menggunakan webcam dengan Raspberry Pi [1,2]. Sistem ini diakses menggunakan smartphone Android dengan media transmisi berupa 3G dongle yang terpasang pada Raspberry Pi. Sang peneliti menggunakan VNC Viewer untuk dapat mengakses ke dalam sistem Raspberry Pi. Kontrol akses pintu dan keamanan rumah menggunakan Raspberry Pi lewat internet. Raspberry Pi model B dapat dimanfaatkan untuk membangun sistem pemantau nirkabel [3]. Dalam penelitiannya menggunakan IP Camera sebagai alat pemantau utama dari sistemnya.

Sistem deteksi slot parkir mobil dapat dibangun menggunakan metode Morfologi dan Background Subtraction [4]. Penelitian dimaksudkan untuk membuat sistem deteksi dan informasi seberapa banyak tersedia lahan parkir mobil melalui modul kamera V2 Raspberry Pi dengan menggunakan metode Background Subtraction dan Morfologi. Sistem keamanan magnetic door lock menggunakan keypad dan solenoid berbasis mikrokontroler Arduino Uno, akan memberikan status bahwa orang yang memasukkan kata sandi tersebut dapat masuk atau tidak dengan mencocokkan kata sandi dengan basis data yang tersimpan di mikrokontroler [5]. Jika kata sandi yang dimasukkan benar maka orang tersebut dapat masuk, jika salah maka LCD penampil akan memberitahukan bahwa kata sandi salah dan orang tersebut tidak dapat masuk ke rumah.

Perbedaan penelitian yang akan dilakukan terdapat penggunaan Raspberry Pi 3 yang dapat dikembangkan dan ditingkatkan lagi sistem keamanannya, serta layanan yang digunakan adalah WLAN. Selain itu penelitian ini menggunakan library OpenCV dan kamera Logitech C270 sebesar 3 megapixel dengan maksimal resolusi mencapai 720p. Metode Haar-Like Cascade Classifier digunakan untuk mengenali wajah manusia secara

*realtime* dan menambahkan sistem notifikasi yang akan dikirimkan ke telepon pintar para penghuni rumah agar mengetahui bahwa ada orang tidak dikenal berada di depan rumah.

### **Raspberry Pi 3 Model B**

Raspberry Pi merupakan sebuah Mini PC dengan harga murah namun berperforma tinggi untuk digunakan orang-orang dalam belajar, memecahkan masalah, dan bersenang-senang. Ukuran Raspberry Pi hanya seukuran tidak lebih besar dari telapak tangan orang dewasa. Sasaran penggunaan Raspberry Pi ini adalah untuk pengembangan project IoT (Internet of Things) dan bersifat Open-source. Raspberry Pi ini menggunakan SoC (System-on-Chip) Broadcom BCM2837 Cortex A53 64-bit CPU dengan 4 inti bertenaga 1,2GHz yang mampu menampilkan gambar di monitor dengan resolusi tertinggi 1080p, GPU Broadcom VideoCore IV dengan frekuensi 400MHz, mendukung teknologi OpenGL ES 2.0, 1GB RAM tipe DDR3, memiliki slot Ethernet RJ45 berkecepatan 100Mbps untuk terhubung dengan internet menggunakan media kabel LAN, mendukung HDMI, 4 slot USB 2.0, sumber daya menggunakan DC input 5 volt minimal 1,5 ampere, GPIO 40 pin multifungsi, dan support banyak sistem operasi seperti Android, Ubuntu, Debian, Raspbian, bahkan KODI. Raspberry Pi 3 Model B memiliki dimensi 86.9mm x 58.5mm x 19.1mm dan berat 41,2g [6].

### **Bahasa Program Python**

Python merupakan bahasa pemrograman tingkat tinggi yang diracik oleh Guido van Rossum [7]. Python banyak digunakan untuk membuat berbagai macam program, seperti: program CLI, Program GUI (desktop), Aplikasi Mobile, Web, IoT, Game, i untuk Hacking, dsb. Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaknya rapi dan mudah dipahami. Python merupakan bahasa pemrograman yang digunakan tidak hanya untuk pemrograman web, Python juga digunakan oleh sebagian besar Distro Linux, IoT Project, dan lain lain. Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya. Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari syntax error.

### **Raspbian**

Raspbian merupakan sistem operasi berbasis Linux Debian Stretch. Raspbian digunakan untuk menjalankan perangkat khususnya Raspberry Pi. Namun, banyak pengembang memodifikasi agar Raspbian ini dapat dijalankan pada beberapa platform. Contohnya adalah Orange Pi PC yang dapat menjalankan Raspbian. Sistem operasi ini sudah terpasang beberapa aplikasi seperti Python IDE, Scratch, Mathematica, dan lain lain.

### **OpenCV**

OpenCV (Open Computer Vision) adalah sebuah API (Application Programming Interface) Library yang sudah sangat familiar pada Pengolahan Citra Computer Vision [8]. Computer Vision itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari Computer Vision adalah Face Recognition,

Face Detection, Face/Object Tracking, Road Tracking, dll. OpenCV adalah library Open Source untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video. OpenCV sendiri terdiri dari 5 library, yaitu: CV (untuk algoritma Image processing dan Vision); ML (untuk machine learning library); Highgui (untuk GUI, Image dan Video I/O); dan CXCORE (untuk struktur data, mendukung XML dan fungsi-fungsi grafis).

### **Metode Haar-cascade Classifier**

OpenCV menggunakan sebuah tipe face detector yang disebut Haar-cascade classifier [9]. Jika ada sebuah image (bisa dari file/live video), face detector akan menguji tiap lokasi image dan mengklasifikasinya sebagai “wajah” atau “bukan wajah”. Klasifikasi wajah ini menggunakan sebuah pemisalan skala yang tetap, misalnya 50×50 pixel. Jika wajah pada image lebih besar atau lebih kecil dari pixel tersebut, classifier terus menerus jalan beberapa kali, untuk mencari wajah pada gambar tersebut. Classifier menggunakan data yang disimpan pada file XML untuk memutuskan bagaimana mengklasifikasi tiap lokasi image. OpenCV menggunakan 4 data XML untuk deteksi wajah depan, dan satu untuk wajah profile. Termasuk juga 3 file XML untuk bukan wajah: satu untuk mendeteksi badan secara penuh, satu untuk badan bagian atas, dan satu untuk badan bagian bawah. Pengguna harus memberitahukan (mendeklarasikan) letak dari classifier yang digunakan. Salah satunya bernama `haarcascade_frontalface_default.xml`.

### **Haar-Like Feature**

Secara umum, Haar-Like Feature digunakan dalam mendeteksi objek pada image digital. Nama Haar merujuk pada suatu fungsi matematika (Haar Wavelet) yang berbentuk kotak, prinsipnya sama seperti pada fungsi Fourier. Awalnya pengolahan gambar hanya dengan melihat dari nilai RGB setiap pixel, namun metoda ini ternyata tidaklah efektif. Paul Viola dan Michael Jones kemudian mengembangkannya sehingga terbentuk Haar-Like feature. Haar-like feature memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa pixel. Per kotak itu kemudian diproses dan didapatkan perbedaan nilai (threshold) yang menandakan daerah gelap dan terang. Nilai-nilai inilah yang nantinya dijadikan dasar dalam image processing. Untuk gambar bergerak (video), perhitungan dan penjumlahan pixel terjadi secara terus-menerus dan membutuhkan waktu yang lama. Oleh karena itu, penjumlahan diganti dengan integral sehingga didapatkan hasil lebih cepat. Hasil deteksi dari Haar-Like kurang akurat jika hanya menggunakan satu fungsi saja sehingga biasanya digunakan beberapa fungsi sekaligus (massal). Semakin banyak fungsi yang digunakan maka hasilnya akan semakin akurat. Pemrosesan Haar-Like feature yang banyak tersebut diorganisir atau diatur di dalam classifier cascade.

### **Konsep Pendeteksian Wajah**

OpenCV face detector menggunakan metode Paul Viola dan Michael Jones yang dipublikasikan pada tahun 2001. Pendekatan ini mendeteksi objek dengan menggabungkan konsep:

1. Fitur rectangular sederhana yang disebut fitur Haar.
2. Integral image untuk deteksi fitur yang cepat.
3. Metode machine learning AdaBoost.

## Push-Safer API

Push-safer merupakan website penyedia layanan untuk mengirim dan menerima push notification kepada perangkat maupun gadget yang terhubung dengan internet, baik perangkat berbasis Android, Windows, Apple, melalui E-mail dan Browser sekalipun. Push-safer menyediakan banyak fitur untuk mendukung pengiriman dan penerimaan notifikasi, dalam bentuk integrasi ke banyak alat, bahasa pemrograman dan aplikasi yang terpasang di telepon pintar maupun plugin browser.

## METODE PENELITIAN

### Alat dan Bahan

Dalam penelitian ini memerlukan *hardware* yang terdiri dari:

1. Raspberry Pi 3
2. Webcam *Logitech C270*
3. Wifi USB Adapter TP-WN725N

Sedangkan sistem operasi yang digunakan adalah *library OpenCV, android, Push-Safer API*.

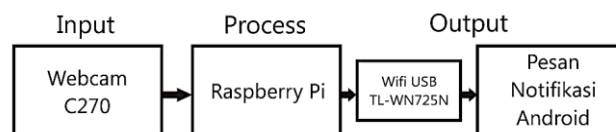
### Diagram Alir Penelitian

Penelitian ini dilakukan dengan beberapa tahapan sebagai berikut:

1. Perencanaan input
2. Perencanaan proses
3. Perencanaan output
4. Perencanaan rangkaian keseluruhan
5. Pengujian

### Blok Diagram

Di bawah ini adalah blok diagram perencanaan dan perancangan sistem keamanan rumah dengan pengenalan wajah menggunakan *webcam* dan *library OpenCV* berbasis *Raspberry Pi*. Menggunakan *webcam* sebagai *input* yaitu *Logitech C270* beresolusi 3 *megapixels*. *Output* alat ini menggunakan *Wifi USB Adapter* yaitu TP-WN725N, dan pengiriman pesan notifikasi ke telepon pintar *Android*.



Gambar 1. Blok Diagram Sistem Keamanan Rumah

### Perencanaan Input

*Webcam Logitech C270* menggunakan lensa beresolusi 3 *megapixel*, memiliki mikrofon, dan maksimal tampilan gambar mencapai 720p atau HD.

### Perencanaan Proses

Pada perancangan proses, menggunakan *Raspberry Pi*. Perangkat *Mini PC* ini mempunyai spesifikasi yang cukup untuk menjalankan sistem keamanan rumah berdasarkan pengenalan wajah dengan menggunakan:

1. Sistem operasi *Raspbian* berbasis *Debian Stretch*.
2. 1 *slot USB* untuk *webcam*
3. *Power DC 5v* dengan arus 3 ampere untuk menyalakan *Raspberry Pi*
4. Slot kartu SD dengan penyimpanan sebesar 16GB

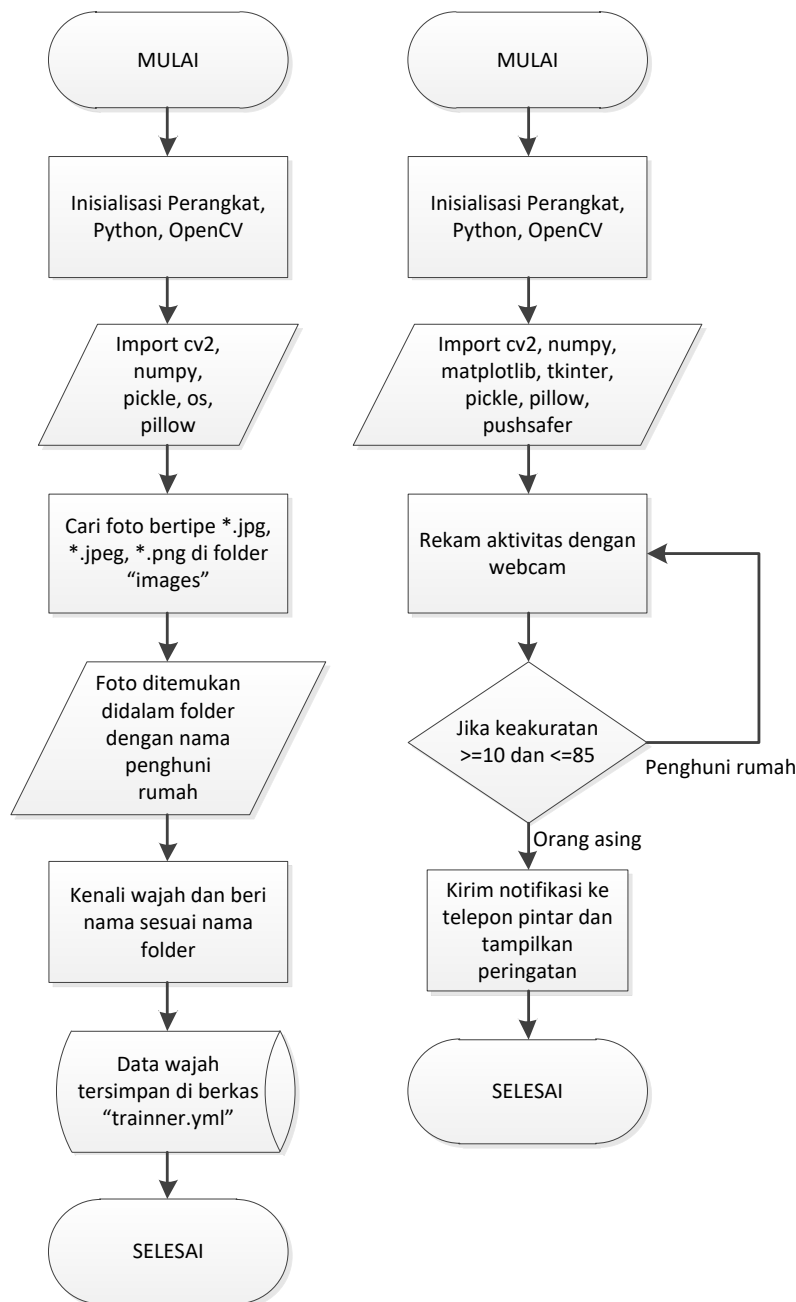
Perancangan proses ini akan mengolah data berupa gambar yang didapat dari *webcam* C270. Data-data yang didapatkan tadi akan diproses di *Mini PC* ini, untuk selanjutnya memberikan *output* ke perangkat *wifi* apakah orang yang berada di depan rumah adalah orang asing atau bukan.

### **Perencanaan Output**

Output menggunakan perangkat TL-WN725N yang menghubungkan *Raspberry Pi* dengan jaringan nirkabel yang ada di rumah. Perangkat *output* ini nantinya yang akan bertugas untuk mengirimkan hasil dari proses pengenalan wajah. Perangkat ini harus terhubung ke *router* atau perangkat apapun yang tersambung dengan akses internet untuk dapat mengirimkan hasil dari proses pengenalan wajah ke telepon pintar dalam bentuk *push notification* yang akan diatur oleh *Push-Safer API*.

### **Perencanaan Program**

Bahasa program yang akan digunakan adalah bahasa pemrograman *Python* versi 3.7, OpenCV dengan modul ekstra “*contrib*”, dan *Push-Safer API*. Sistem keamanan rumah akan menggunakan fitur yang bernama *Haar-cascade* untuk mengenali setiap wajah yang terekam di *webcam*, apakah yang terekam adalah penghuni rumah atau bukan.



Gambar 2. Flowchart Program Sistem Keamanan Rumah

Informasi yang dikirimkan berupa *push notification*, berisi informasi baik berupa teks. *Push-safer* akan digunakan untuk mengirimkan informasi hasil pengolahan data yang dilakukan sebelumnya oleh *Raspberry Pi* berupa notifikasi yang bukan penghuni rumah. Nantinya, notifikasi yang akan dikirimkan akan berjudul “Peringatan!”, berisi “Ada orang asing berada di depan rumah!”. *Push-safer* diakses melalui halaman web yang beralamatkan <https://pushsafer.com>. Untuk menggunakan *Push-safer*, login terlebih dahulu dengan memasukkan *email* dan *password* yang sudah didaftarkan. Setelah *login*, pengguna akan masuk di tampilan *dashboard* yang berisi beberapa informasi seperti kode unik yang akan digunakan untuk memanggil API dari *Push-safer*, perangkat yang terdaftar, kuota untuk memanggil API *Push-safer*, parameter yang digunakan, mengirim notifikasi instan, dan lain-lain.

Metode pengenalan wajah menggunakan *Haar-cascade* dan dideklarasikan dalam string *faceCascade*. Untuk fungsi pengenalan wajah, dideklarasikan dalam string *recognizer*. Dalam pengenalan wajah, sistem perlu membandingkan antara wajah yang terekam oleh sistem dan wajah yang ada di basis data, dideklarasikan dalam perintah *recognizer.read("trainer.yml")*. Untuk isi pesan peringatan baik di sistem maupun *push notification*, dideklarasikan dalam string *title\_orangasing = "Peringatan!"* dan *body\_orangasing = "Ada orang asing berada di depan rumah!"*. Deklarasi terakhir adalah *init("<Masukkan API Key Disini>")*, *push-safer* menggunakan *API key* yang perlu dipanggil terlebih dahulu agar notifikasi dapat dikirimkan. *API key* tersedia saat pengguna *login* di halaman *web push-safer* pada bagian *dashboard*. Nama pemilik wajah perlu ditampilkan untuk memberitahukan siapa pemilik wajah yang terekam oleh sistem.

### Pengujian

Sebelum diuji, wajah dari para penghuni rumah perlu dimuat terlebih dahulu dengan file *faces-train.py* yang nantinya akan membentuk basis data bernama *trainer.yml*, file ini digunakan sebagai acuan perbandingan untuk pendeteksian antara basis data yang tersimpan dengan wajah yang tertangkap dari *webcam*. Wajah yang akan di-*training* diperoleh dari beberapa foto wajah para penghuni rumah. Foto yang digunakan adalah foto wajah dengan berbagai ekspresi sehingga sistem dapat mengenali wajah secara baik. Foto yang digunakan antara 8-10 foto dengan format *.jpg*, *.png* dan *.jpeg*.



Gambar 3. Foto Wajah Penghuni Rumah “audina”

Foto wajah disimpan di folder dengan nama sesuai dengan nama penghuni rumah, begitupun seterusnya saat ada penghuni rumah yang ingin dimasukkan basis data wajahnya. Untuk men-*train* wajah, menggunakan file *faces-train.py*, untuk mengeksekusinya menggunakan perintah *python3 faces-train.py* di *terminal emulator Linux*. Hasilnya adalah nilai dengan tipe data *uint8* dan angka koordinat X, Y, W, H wajah dari setiap foto yang terlihat pada *terminal emulator*. Semua data mulai dari foto dan koordinat wajah sudah otomatis masuk kedalam basisdata bernama “*trainer.yml*” pada saat *faces-train.py* selesai dijalankan. Data bertipe *uint8* inilah yang akan digunakan sebagai acuan basisdata untuk pengenalan wajah dengan membandingkan antara wajah yang tertangkap dengan basisdata yang ada.



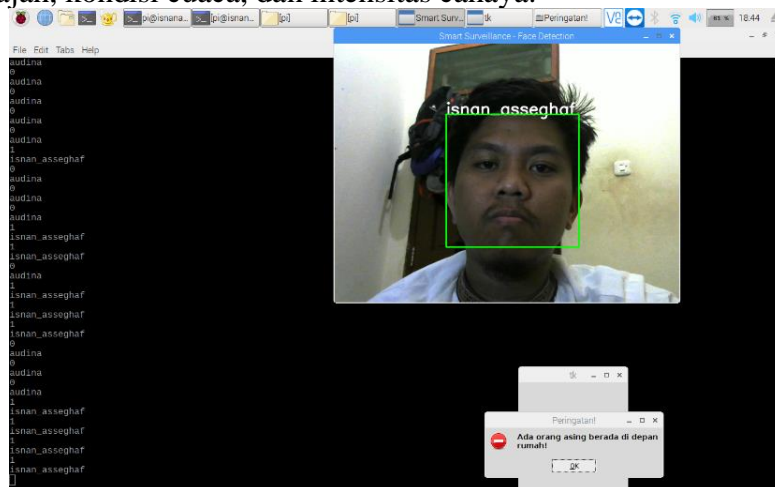
```
okiputani E:\NEW_Recognizer\images\okiputani\6.jpeg  
[[ 59 64 66 ... 143 141 140]  
[ 59 63 66 ... 150 148 147]  
[ 69 63 65 ... 157 154 153]  
...  
[ 71 71 71 ... 41 39 39]  
[ 71 71 71 ... 39 38 37]  
[ 71 71 71 ... 39 37 37]  
191 268 403 403  
okiputani E:\NEW_Recognizer\images\okiputani\7.jpeg  
[[ 24 24 24 ... 148 148 148]  
[ 24 24 24 ... 148 148 148]  
[ 24 24 24 ... 148 148 148]  
...  
[ 9 9 9 ... 14 14 14]  
[ 10 9 8 ... 14 14 14]  
[ 10 9 8 ... 14 14 14]  
424 431 298 298  
okiputani E:\NEW_Recognizer\images\okiputani\8.jpeg  
[[101 63 82 ... 211 254 244]  
[ 88 73 95 ... 247 247 253]  
[ 93 99 116 ... 253 229 241]  
...  
[ 37 35 32 ... 36 36 36]  
[ 37 35 32 ... 36 36 36]  
[ 37 35 32 ... 36 36 36]  
40 70 122 122  
okiputani E:\NEW_Recognizer\images\okiputani\9.jpeg  
[[237 237 237 ... 239 239 239]  
[237 237 237 ... 240 239 239]  
[237 237 237 ... 240 240 239]  
...  
[ 13 12 10 ... 30 30 31]  
[ 13 12 10 ... 30 31 31]  
[ 13 12 10 ... 30 31 31]  
276 272 486 486
```

Gambar 4. *Output* Data Wajah yang Sudah Masuk Ke dalam Basis Data

Semua pengujian akan dilakukan oleh penghuni rumah “isnan\_asseghaf” dan beberapa orang asing untuk menguji sistem keamanan.

### Pengujian Hasil Pengenalan Wajah

Pengujian hasil pendeteksian dilakukan dengan cara membandingkan wajah yang tertangkap di kamera dengan wajah yang ada pada basis data. Hasil dari pendeteksian tersebut kemudian dilakukan analisis, baik tingkat keakuratan, sudut kemiringan wajah dan keterbacaan wajah, kondisi cuaca, dan intensitas cahaya.



Gambar 5. Pengujian Pendeteksian Wajah

### Pengujian Pengiriman Notifikasi ke Telepon Pintar

Sebelum diuji dengan sistem, terlebih dahulu diuji dengan mengirimkan notifikasi *manual* dari *website* halaman *push-safer* ke telepon pintar yang sudah terpasang aplikasi *client push-safer*. Pengujian ini bertujuan untuk mengetes apakah notifikasi terkirim dengan baik atau tidak. Sebagai catatan, faktor kecepatan internet yang digunakan mempengaruhi kecepatan pengiriman notifikasi ke telepon pintar.

Selanjutnya, pengujian pengiriman notifikasi dilakukan dengan cara mengukur kecepatan pengiriman notifikasi kepada telepon pintar penghuni rumah ketika terdeteksi wajah yang bukan penghuni rumah. Pengujian akan dilakukan dengan menghadapkan *webcam* ke wajah penghuni rumah dan yang bukan penghuni rumah. Hal ini bertujuan untuk mengukur seberapa cepat pemberitahuan dari sistem ke telepon pintar penghuni rumah.

## HASIL DAN PEMBAHASAN

### Hasil Pengujian Pendeteksian Sistem

Semua parameter pengujian dilakukan pada siang hari, kecuali pendeteksian malam hari hanya untuk memastikan sistem mendeteksi wajah apa tidak, dikarenakan ada keterbatasan pengujian pada saat malam hari. Pada tabel 1, dapat diketahui bahwa sistem dapat mendeteksi wajah dengan jarak tertentu. Hasil pengujian berpengaruh pada faktor posisi kamera, waktu, dan posisi cahaya matahari. Pada pengujian dengan jarak tertentu, penghuni rumah dianggap orang asing. Tes beberapa kali dilakukan untuk memastikan bahwa sistem bekerja.

Tabel 1. Pendeteksian Wajah Berdasarkan Jarak Antara Webcam dan Wajah

Jarak webcam dan wajah	Hasil pendeteksian
50cm	Terdeteksi
100cm	Tidak terdeteksi
150cm	Tidak terdeteksi
200cm	Terdeteksi

Tabel selanjutnya adalah pengujian pendeteksian berdasarkan waktu, yaitu siang hari dan malam hari.

Tabel 2. Pendeteksian Wajah Berdasarkan Waktu

Waktu	Hasil pendeteksian
Siang (14.18-14.50)	Terdeteksi
Malam (23.20-23.30)	Terdeteksi

Sistem diuji coba dengan beberapa orang menghadap ke depan webcam.

Tabel 3. Pengujian dengan Beberapa Orang Tertangkap Webcam

Jumlah orang di webcam	Hasil pendeteksian
1	Terdeteksi
2	Terdeteksi (hanya satu orang)
3	Terdeteksi (hanya satu orang)

Sistem juga diuji dengan skenario baik penghuni rumah maupun orang asing berjalan dari gerbang depan menuju pintu rumah.

Tabel 4. Pengujian dengan Berjalan dari Gerbang Depan Menuju Pintu Rumah

Orang	Hasil pendeteksian
Penghuni rumah	Terdeteksi (Orang asing)
Penghuni rumah	Terdeteksi (Orang asing)
Penghuni rumah	Terdeteksi (Penghuni rumah, Orang asing)
Orang asing	Terdeteksi (Orang asing)

### Hasil Pengujian Pengiriman Notifikasi

Hasil dari pengujian ini adalah apakah notifikasi terkirim dengan baik atau tidak setelah mendeteksi orang asing.

Tabel 5. Status Pengiriman Notifikasi Orang Asing

Deteksi ke	Status notifikasi
------------	-------------------

1	Terkirim
2	Trrkirim
3	Terkirim

### Analisis Hasil

#### Berdasarkan Jarak Webcam dan Wajah

Jarak antara webcam dan wajah dengan *interval* 50cm sampai pada jarak 200cm. Waktu pengujian adalah 2 menit 38 detik. Pada tabel 6 menunjukkan hasil pendeteksian dengan jarak antara webcam dan wajah.

Tabel 6. Hasil Deteksi Berdasarkan Jarak dan Wajah

Jarak	Orang Sebenarnya	Hasil Deteksi (Banyaknya Deteksi)
50cm	Isnan_asseghaf	Isnan_asseghaf (46), Orang_asing (1)
100cm	Isnan_asseghaf	Tidak terdeteksi
150cm	Isnan_asseghaf	Tidak terdeteksi
200cm	Isnan_asseghaf	Orang_asing (24)
<b>Total deteksi</b>		<b>71</b>

Total keseluruhan deteksi:

$$\frac{46}{71} \times 100 = 64,8 \% \text{ (terdeteksi benar)}$$

$$\frac{25}{71} \times 100 = 35,2 \% \text{ (terdeteksi salah)}$$

#### Siang dan Malam

Hasil pengujian berdasarkan waktu, yaitu siang dan malam. Tabel 7 menunjukkan hasil pendeteksian siang dan malam.

Tabel 7. Hasil Deteksi Berdasarkan Waktu

Waktu	Orang sebenarnya	Hasil deteksi (banyaknya deteksi)
Siang (14.18-14.50)	Isnan_asseghaf	Isnan_asseghaf (46), Orang_asing (25)
Malam (23.20-23.30)	Isnan_asseghaf	Isnan_asseghaf (10), Audina (11), Orang_asing (18)
<b>Total deteksi</b>		<b>110</b>

Untuk perhitungan persentase kesalahan deteksi dan total keseluruhan kesalahan deteksi pada waktu siang dan malam adalah sebagai berikut.

Persentase kesalahan deteksi siang hari:

$$\frac{25}{71} \times 100 = 35,2 \% \text{ (kesalahan deteksi)}$$

Persentase kesalahan deteksi malam hari:

$$\frac{29}{39} \times 100 = 74,3 \% \text{ (kesalahan deteksi)}$$

Persentase total kesalahan:

$$\frac{25}{110} \times 100 = 49,1 \% \text{ (kesalahan deteksi)}$$

#### Berdasarkan Banyaknya Orang

Sistem diuji dengan banyaknya orang yang tertangkap di webcam. Sistem diuji dengan mengenali wajah sampai dengan 3 orang. Waktu pengujian adalah 2 menit 35 detik. Hasil pengujiannya dapat dilihat pada tabel 8.

Tabel 8. Hasil Deteksi Berdasarkan Banyaknya Orang

Banyaknya orang	Jumlah terdeteksi	Hasil deteksi (Banyaknya deteksi)
Isnan_asseghaf	1	isnan_asseghaf (8), orang_asing (2)
Isnan_aseghaf, Orang_asing	1	Orang_asing (2), Orang_asing (5)
Isnan_asseghaf, Orang_asing, Orang_asing	1	Isnan_asseghaf (23), Okiputami (1), Tidak terdeteksi, Orang_asing (7)
<b>Total Deteksi</b>		<b>48</b>

Persentase kesalahan deteksi untuk satu orang:

$$\frac{2}{(8 + 2)} \times 100 = 20 \% \text{ (kesalahan deteksi)}$$

Persentase kesalahan deteksi untuk dua orang:

$$\frac{2}{(2 + 5)} \times 100 = 28,5 \% \text{ (kesalahan deteksi)}$$

Persentase kesalahan deteksi untuk tiga orang:

$$\frac{1}{(23 + 1 + 7)} \times 100 = 3,2 \% \text{ (kesalahan deteksi)}$$

Persentase total kesalahan deteksi berdasarkan banyaknya orang satu sampai tiga orang di depan kamera:

$$\frac{(2 + 2 + 1)}{48} \times 100 = 10,4 \% \text{ (kesalahan deteksi)}$$

### Skenario Orang Berjalan Dari Gerbang ke Pintu Rumah

Pengujian ini dilakukan untuk menguji seberapa baik sistem keamanan saat mengenali wajah pada saat orang bergerak dari gerbang menuju pintu rumah.

Tabel 9. Hasil Deteksi Berdasarkan Skenario Orang Berjalan

Percobaan ke	Orang sebenarnya	Hasil deteksi (Banyaknya deteksi)
1	Isnan_asseghaf	Orang_asing (3)
2	Isnan_asseghaf	Orang_asing (6)
3	Isnan_asseghaf	Orang_asing (3), Isnan_asseghaf (8)
4	Orang_asing	Orang_asing (6)
<b>Total deteksi</b>		<b>26</b>

Persentase kesalahan deteksi percobaan ke satu:

$$\frac{3}{3} \times 100 = 100 \% \text{ (kesalahan deteksi)}$$

Persentase kesalahan deteksi percobaan ke dua:

$$\frac{6}{6} \times 100 = 100 \% \text{ (kesalahan deteksi)}$$

Persentase kesalahan deteksi percobaan ke tiga:

$$\frac{3}{11} \times 100 = 27,3 \% \text{ (kesalahan deteksi)}$$

Persentase total kesalahan berdasarkan skenario orang berjalan dari gerbang ke pintu rumah:

$$\frac{(3 + 6 + 3)}{26} \times 100 = 46,1 \% \text{ (kesalahan deteksi)}$$

### Pengiriman notifikasi ke telepon pintar

Pada dasarnya sistem ini difungsikan untuk mendeteksi adanya orang asing yang datang ke rumah, oleh karena itu perlu dilakukan pengujian pengiriman notifikasi ke telepon pintar.

Tabel 4.10 Hasil Pengiriman Notifikasi ke Telepon Pintar

Deteksi ke	Status notifikasi
1	Terkirim
2	Terkirim
3	Terkirim
<b>Total terkirim</b>	<b>3</b>

### KESIMPULAN

Berdasarkan hasil pengujian dan analisis, dapat ditarik kesimpulan sebagai berikut:

1. Sistem keamanan rumah dirancang dengan menggunakan beberapa perangkat keras yaitu *Raspberry Pi 3 Model B*, modul *Webcam Logitech C270* beresolusi *3 Megapixel*, *TP-Link USB Wifi Adapter* dan untuk perangkat lunak menggunakan *library OpenCV* yang ditulis dalam bahasa *Python* versi 3 dengan menggunakan metode *Haar-cascade Classsifier* untuk pengenalan wajah serta *Push-Safer* sebagai penyedia layanan notifikasi untuk telepon pintar berbasis *Android*.
2. Saat sistem keamanan rumah mengenali wajah orang yang bukan penghuni rumah, sistem akan mengirimkan pesan notifikasi menggunakan layanan notifikasi dari *Push-Safer* melalui koneksi internet.
3. Dengan menggunakan *Raspberry Pi 3*, modul *Webcam Logitech C270* beresolusi *3 Megapixel*, *TP-Link Wifi USB Adapter*, *Python 3*, *Library OpenCV*, *Push-Safer* dan metode *Haar-cascade Classifier*, berdasarkan hasil pengujian dan analisis didapatkan persentase total kesalahan pendeteksian berdasarkan jarak sebesar 35,2%, total kesalahan deteksi berdasarkan banyaknya orang dengan satu sampai tiga orang sebesar 10,4 % dan total kesalahan deteksi berdasarkan skenario orang berjalan dari gerbang ke pintu rumah sebesar 46,1 %. Persentase kesalahan untuk siang hari sebesar 35,2 %, untuk malam hari sebesar 74,3 %.

### SARAN

Berdasarkan hasil penelitian dan pengujian, dapat diberikan saran-saran yang bertujuan meningkatkan kinerja dan kualitas sistem keamanan ini, adalah sebagai berikut:

1. Foto wajah ditambahkan lebih dari 10 dengan kualitas resolusi yang cukup tinggi, bermacam-macam ekspresi serta sudut wajah yang bervariasi sehingga akurasi pengenalan wajah dapat ditingkatkan dan mengurangi kesalahan pembacaan.
2. Karena keterbatasan pemanggilan *API* untuk layanan *Push-notification* dari *Push-safer*, disarankan mengganti layanan *push-notification* tanpa batasan pemanggilan *API* sehingga layanan notifikasi dapat digunakan semaksimal mungkin tanpa perlu khawatir kehabisan *API calls* di saat sistem keamanan rumah sedang bekerja.

### DAFTAR PUSTAKA

- [1] Swathi, R., et al. (2014). "Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor." *International Journal of Computer Science and Information Technology (IJCSIT)* Vol. 5 No. 6., 7107-7109.
- [2] Sarker, Srijon., Nooman, Md. Shiblee., dan Chowdhury, Md. Nasimuzzaman (2013). "Access Control of Door and Home Security by Raspberry Pi Through Internet." *International Journal of Scientific & Engineering Research* Vol. 4 No. 11., 550-558.
- [3] Darjat, Sudjadi, Shadiq, dan Helmi Muhammad (2014). "Perancangan Kamera Pemantau Nirkabel Menggunakan Raspberry PI Model B." *TRANSIENT* Vol. 3 No. 4., 546-551.
- [4] Maulana, Rizqy., Fitriyah, Hurriyatul., Prakasa, Esa. (2018). "Implementasi Sistem Deteksi Slot Parkir Mobil Menggunakan Metode Morfologi dan Background Subtraction." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* Vol. 2, No. 5, 1954-1959.
- [5] Guntoro, Helmi., dkk. (2013). "Rancang Bangun Magnetic Door Lock Menggunakan Keypad dan Solenoid Berbasis Mikrokontroler Arduino Uno." *Jurnal ELECTRANS* Vol. 12 No.1., 39-48.
- [6] Fernando, Erick (2014). "Automatisasi Smart Home dengan Raspberry PI dan Smartphone Android." *Konferensi Nasional Ilmu Komputer (KONIK)* 1-5.
- [7] Dian, Muhar (2014). *Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal*. [Online]. Tersedia: <https://www.petanikode.com/python-linux/> [30 Agustus 2018].
- [8] Priawadi, Ozi. (2012). *OpenCV* [Online]. Tersedia: <http://www.priawadi.com/2012/09/opencv.html> [15 April 2018].
- [9] Sajati, Haruno. (2015). *Deteksi Obyek Menggunakan Haar Cascade Classifier* [Online]. Tersedia: <http://jati.stta.ac.id/2015/09/deteksi-obyek-menggunakan-haar-cascade.html> [15 Juli 2018].