Rancang Bangun Sistem *Surveillance* untuk Keamanan Pintu Rumah dengan Metode PCA- Haar Cascades Berbasis *Internet of Things*

Dimas Waluyo Jati
dmswjt@gmail.com
Universitas Jember

Catur Suko Sarwono
catursuko@gmail.com
Universitas Jember

Widya Cahyadi
cahyadi@unej.ac.id
Universitas Jember

Abstrak

Peningkatan jumlah dan minimnya tingkat penyelesaian kasus kejahatan rumah tangga perlu untuk segera ditangani. problematika ini bisa ditanggulangi dengan menggunakan sistem keamanan berupa sistem pengawas yang dilengkapi dengan notifikasi melalui aplikasi *chatting* dan sistem penyimpanan *cloud storage* sehingga dapat diakses di manapun. Penelitian ini fokus pada rancang bangun algoritma *Principal Component Analysis* (PCA) pada sistem kamera pengawas atau *surveillance* yang berbasis *Internet of Things* dan keamanan pintu rumah dengan menggunakan Raspberry Pi dengan pustaka *Open Source Computer Vision* (OpenCV). Konsep dari rancang bangun sistem ini adalah, saat sistem mendeteksi gerakan, sistem akan *capture* pergerakan subjek dan sistem akan mengunggah hasil *capture* tersebut ke akun Dropbox dari pemilik rumah, dan disaat yang bersamaan, sistem akan memberi notifikasi ke akun Telegram pemilik rumah. Sistem pengenalan wajah menggunakan metode PCA mempunyai keakuratan data sebesar 93,75%, dengan nilai *error* sebesar 6,25%.

Kata Kunci — **PCA, *Surveillance*, Internet of Things.**

Abstract

An increase in the number and lack of the resolution household crimes need to be immediately treated. To cope with the problems are required the security system of monitoring system equipped with notifications through the application of chat and system storage cloud storage and anywhere in access. This study focused on designed up algorithms *Principal Component Analysis* (PCA) on camera system superintendent or surveillance based on the internet of things and security the home using raspberry pi with *Open Source Computer Vision Library* (OpenCV) library. The concept of designed up this system is, when the system detect movement, systems to capture the subject and systems to upload the capture to the accounts dropbox from the homeowner, and at the same time, the system will give notification telegram into account the homeowner. A facial recognition using the pca method have data accuracy of 93,75 %, with value of error is 6,25 %.

Keywords **— PCA, *Surveillance*, Internet of Things.**

I. PENDAHULUAN

Jumlah rumah tangga yang pernah menjadi korban kejahatan cenderung meningkat sedangkan persentase penyelesaian tindak pidana oleh Polri hanya sebesar 58,13%. Untuk menanggulangi problematika ini diperlukan sistem keamanan berupa sistem pengawas atau *surveillance*, dan untuk memudahkan penghuni rumah, sistem *surveillance* tersebut dapat dilengkapi dengan notifikasi melalui aplikasi *chatting* dan sistem penyimpanan *cloud storage* sehingga dapat diakses di manapun. Selain itu diperlukan juga sistem keamanan yang lebih cerdas dibandingkan pengaman konvensional pada pintu rumah. Misal dengan menambakan sistem autentikasi. Salah satu contoh sistem autentikasi yang sering digunakan adalah teknik identifikasi biometrik menggunakan wajah atau *face recognition*.

Penelitian ini fokus pada rancang bangun algoritma *Principal Component Analysis* (PCA) pada sistem kamera pengawas atau *surveillance* yang berbasis *Internet of Things* dan keamanan pintu rumah dengan menggunakan Raspberry Pi dengan pustaka *Open Source Computer Vision* (OpenCV) dan mengetahui tingkat akurasi pengenalan wajah *frontal faces* pada algoritma PCA dengan kondisi cahaya yang berbeda (dalam persen). PCA adalah teknik standar yang digunakan dalam pengenalan pola statistikal dan pemrosesan sinyal untuk data *reduction* dan ekstraksi fitur.[1] *Internet of Things*, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus.[2].

II. STUDI PUSTAKA

Surveillance System ditujukan untuk memonitor berbagai aktivitas di suatu lokasi dengan menggunakan sensor visual. Untuk kebutuhan yang lebih spesifik, sistem ini digunakan untuk keperluan deteksi, pengenalan, dan identifikasi obyek yang di monitor.[3].

Wajah manusia memainkan peran penting dalam interaksi sosial kita, menyampaikan identitas masyarakat. Identifikasi wajah (*face recognition*) merupakan identifikasi biometrik. Biometrik maksudnya pengenalannya berdasarkan keunikan

seseorang atau keadaan fisik maupun karakteristiknya. Ada tiga fase yang diperlukan mesin untuk mengenali wajah. Ketiga fase tersebut adalah *detection*, *feature extraction*, *recognition*[4].

OpenCV menggunakan algoritma pembelajaran oleh mesin untuk mencari wajah dalam sebuah gambar. Algoritma memecah-mecah tugas untuk mengidentifikasi wajah menjadi ribuan tugas yang lebih kecil, seukuran petak kecil, yang mana setiap petaknya mudah untuk dipecahkan. Tugas tersebut juga dinamakan *classifier*. OpenCV menggunakan *cascade* dalam mendeteksi wajah. Keunggulannya adalah mayoritas dari gambar yang tidak memiliki wajah akan terdeteksi negatif selama beberapa tahapan awal, yang mana berarti algoritmanya tidak akan membuang waktu untuk menguji semua fitur pada gambar. Oleh sebab itu, ketimbang memerlukan berjaman, pendeteksian wajah kini dapat dilakukan secara *real time*[5].

PCA memberikan transformasi ortogonal yang disebut juga dengan *eigenimage* dimana sebuah citra akan direpresentasikan ke dalam bentuk proyeksi linier searah dengan *eigenimage* yang bersesuaian dengan nilai Eigen terbesar dari matriks *covariance*. Dalam prakteknya, matriks *covariance* ini dibangun dari sekumpulan image training yang diambil dari berbagai objek.

Misalnya ada sejumlah N individu yang dijadikan sampel. Dari setiap individu diambil P citra, sehingga total citra di dalam *training* set adalah :

$$M = N \times P \quad (1)$$

Sejumlah M sampel citra dinyatakan sebagai {Γ1, Γ2, Γ3, … Γm} di dalam sebuah ruang citra n2 dimensi. Kumpulan citra tersebut dihitung nilai rata-ratanya yang disebut juga sebagai *average face* dengan perhitungan berikut:

$$\psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma_n \quad (2)$$

Setiap citra wajah dikurangi dengan nilai rata-rata membentuk kumpulan vektor menggunakan fungsi:

$$\Phi_i = \Gamma_i - \Psi \quad (3)$$

Kumpulan vektor yang sangat besar ini kemudian mengikuti pada aturan PCA, yang mencari sejumlah M vektor-vektor ortonormal Uk dan nilai Eigen λk yang terbaik dalam menggambarkan distribusi dari data tersebut. Vektor-vektor Uk dan nilai-nilai λk adalah vektor-vektor Eigen dan nilai-nilai Eigen dari matriks *covariance*.

$$C = \frac{1}{M}\sum_{i=1}^{M} \Phi i \Phi i^T = A\,A^T \quad (4)$$

Dimana $A = (\phi_1 \phi_2 \ldots \phi_M)$

Dari matriks C (dalam penelitian ini berukuran n2×n2), diambil vektor-vektor Eigen terbaik sebanyak jumlah data. Karena vektor-vektor Eigen ini memiliki dimensi yang sama dengan dimensi citra yang asli, maka vektor-vektor ini jika disusun menjadi matriks berukuran n×n akan membentuk citra yang mirip dengan wajah. Oleh karena itu, vektor-vektor Eigen ini disebut juga Eigenfaces.

III. METODE

Prototipe ini dibagi ke dalam dua bagian, yaitu, bagian *hardware* dan bagian *software*. Secara umum *hardware* sistem ini terdiri atas tiga bagian utama. Tiga bagian tersebut adalah kamera (a), Raspberry-Pi sebagai *controller* (b) dan sistem kunci (c).

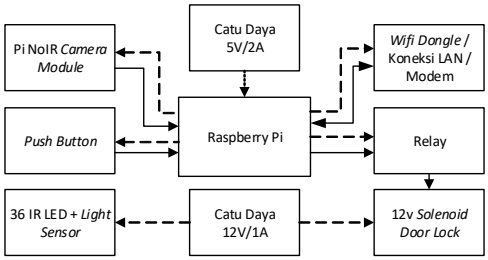
A. Perancangan Perangkat Keras

Pada rangkaian Gbr. 1 terdapat dua alat yang akan dikendalikan oleh raspberry-pi, yaitu: alat pengunci pintu (*relay* dan *solenoid*), dan Pi-NoIR camera. IR-LED *Illuminator* sudah terintegrasi dengan *light* sensor. *Infrared* LED *Illuminator* diparalel dengan *solenoid* untuk mendapatkan sumber tegangan sebesar 12 volt 1 ampere dari *power supply*. *Light* sensor digunakan agar IR-LED *Illuminator* dapat menyala secara otomatis dalam keadaan gelap. menggunakan *Infrared* LED untuk membantu pencahayaan Pi-NoIR camera yang mampu mendeteksi cahaya *infrared* sehingga sistem tetap dapat bekerja dalam keadaan gelap.

Program *surveillance* dimulai dari proses authorisasi dropbox *upload*er. sistem memastikan apakah sistem sudah diizinkan untuk mengakses akun dropbox atau belum. Setelah pengguna menekan bel untuk menginisialisasi kamera, wajah subjek yang telah ditangkap akan memasuki tahap *pre-processing*, yaitu meliputi merubah citra menjadi citra *grayscale, crop, resize* setelah itu diproses menggunakan algoritma PCA (untuk dicocokkan dengan wajah yang ada dalam database) Proses inisialisasi average background meliputi menghitung rata-rata dari *previous frame* dan *current frame*. *Contour detection* adalah nama lain dari *edge detection*, yaitu operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Proses ini berfungsi sebagai penanda, bahwa adanya gerakan pada *frame* yang melebihi *minimum area*. Jika *contour* lebih besar dari *minimum area*, maka status sistem akan berubah menjadi "Ada Gerakan!" dan sistem akan melakukan *capture* pada current *frame*. Disaat yang bersamaan sistem akan mengirim notifikasi ke Telegram *user* dan mengunggah hasil *capture* ke Dropbox..

B. Perancangan Sistem Kunci dan Pengenalan Wajah

Setelah semua syarat terpenuhi, *hardware* sudah terpasang dan *software* sudah di-*install*. Terdapat dua tahap lagi yaitu tahap *Training* dan Uji Klasifikasi. Set untuk data *training* dengan gambar wajah yang diizinkan atau *positive training images* diperoleh dengan menangkap wajah *user* atau pemilik rumah (registrasi). Program *training* dimulai dari membaca *database positive dataset* dan *negative dataset*. Setelah semua



Gbr 1. Diagram blok perangkat keras.

gambar yang akan di-*training* sudah terbaca, akan dibuat model Eigenfaces. Program ini dimulai dari membuat model *Eigenfaces Recognizer*. setelah data siap, sistem akan menginisialisasi kamera dilanjutkan dengan inisialisasi kondisi *push button*. Disaat yang bersamaan dengan *push button* ditekan, sistem melakukan inisialisasi sistem kunci yang diwakili oleh *relay*. Setelah gambar diperoleh, sistem melakukan tahap *pre-processing* meliputi merubah gambar mencadi citra *grayscale*. Lalu sistem mencari wajah pada gambar dengan algoritma haar-cascades.Setelah ditemukan wajah pada gambar, program akan melakukan *crop* pada wajah yang terdeteksi. Setelah gambar di-*crop* pada bagian wajah, program melakukan *resize* dengan ukuran 92 x 112 pixel. selanjutnya adalah memproyeksikan fitur wajah tersebut dengan model *Eigenfaces recognizer* dari wajah *training* yang sudah dibuat di awal program, disaat yang bersamaan program memberikan label terhadap wajah tersebut berdasarkan nilai *confidence value*. Dalam sistem ini hanya terdapat dua label, label "1" disebut "*Positive* Label", label ini merupakan label yang diberikan kepada wajah yang dikenali dan diizinkan untuk memasuki ruangan. Label "2" disebut dengan "*Negative* Label".

IV. Hasil dan Pembahasan

A. *Pengujian Sistem Pengunci Pintu*

Hasil ketika program penguji sistem pengunci pintu ketika dijalankan adalah program tersebut akan memberikan sinyal *low* atau 0 pada GPIO.22 untuk menyalakan *solenoid*, kemudian jeda selama 0,2 detik, lalu memberikan sinyal *high* atau 1 pada GPIO.22 untuk mematikan *solenoid*. Pada Gbr 2. (a) menunjukkan saat *relay* mendapatkan sinyal *low* dari raspberry, tegangan sebesar 5 V dikirimkan dari raspberry menuju *relay* module, hal ini ditandai dengan menyalanya LED warna merah. Kemudian saklar aktif dan arus listrik mengalir melalui *common* ke terminal NO (*Normally Open*) menuju ke *solenoid*.

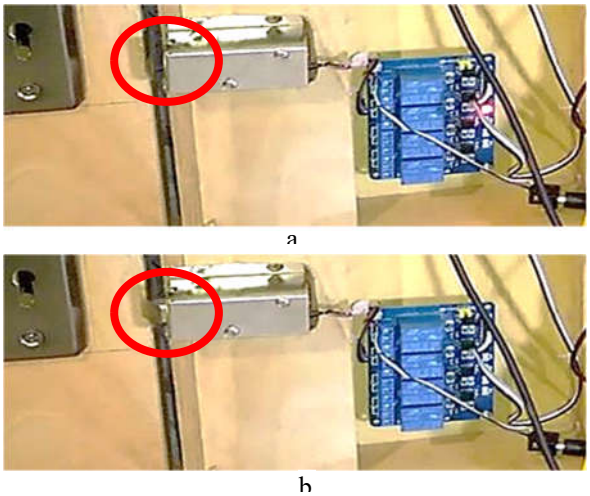
Arus yang mengalir pada kumparan yang terdapat di *solenoid* tersebut menghasilkan gaya elektromagnet yang akan menarik inti besi beserta tuas pengunci ke dalam dan pintu dapat dibuka. Pada Gbr 2. (b) menunjukkan kondisi saat *relay* dan *solenoid* tidak memperoleh sinyal apapun dari raspberry (kondisi *default* pengunci pintu) atau saat *relay* mendapatkan sinyal *high* dari raspberry, tegangan sebesar 5 V yang sebelumnya dikirimkan dari raspberry menuju *relay module* dihentikan, hal ini ditandai dengan padamnya LED warna merah yang sebelumnya menyala. Kemudian, tuas saklar yang sebelumnya mengalirkan arus listrik dari *common* menuju ke terminal NO (ke *solenoid*), kini mengalir dari melalui common menuju ke terminal NC (*Normally Closed*).

Inti besi beserta tuas pengunci yang sebelumnya ditarik oleh gaya elektromagnet, kini terlepas, dan kembali dalam keadaan normal. Saat tuas pengunci yang terdapat pada *solenoid* dalam keadaan normal, pintu tidak dapat dibuka.

B. *Pengujian Telegram*

Pengujian dilakukan dengan mengirimkan pesan dari aplikasi Telegram pada *smartphone* menuju ke kontak "Rumah Dimas" yang sudah teregistrasi dan terdapat pada Telegram API (pada raspberry), kemudian program BOT Telegram API (pada raspberry) mengirimkan pesan balasan ke kontak pemilik rumah, dalam pengujian ini kontak pemilik rumah yang terdapat pada *smartphone* terdaftar atas nama "Dimas Waluyo Jati".

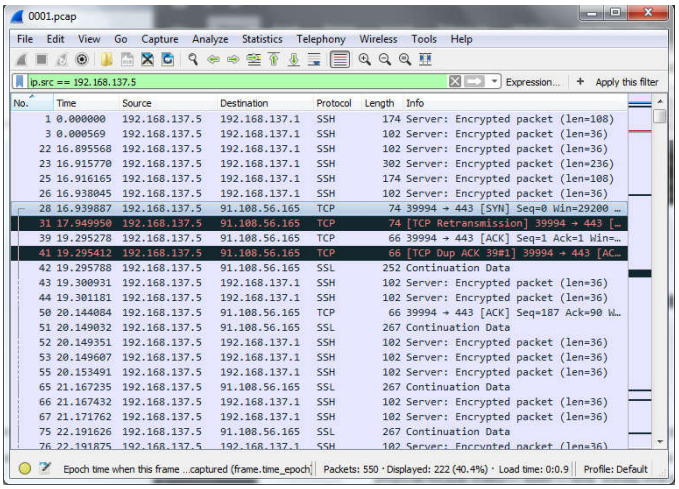
Dapat dilihat pada Gbr. 3, IP 192.168.137.5 atau raspberry hanya berkomunikasi dengan dua IP *address*, yang pertama adalah 192.168.137.1 dan yang kedua adalah 91.108.56.165. IP 192.168.137.1 adalah IP *address* dari PC yang menjadi SSH client. IP 91.108.56.165 merupakan IP dari Telegram. Untuk meyakinkan pendapat bahwa IP tersebut merupakan IP dari server Telegram, penulis melakukan tracert dan *who-is* ke IP 91.108.56.165 dengan menggunakan program *Visual Route* 2010 yang sudah terpasang pada PC.



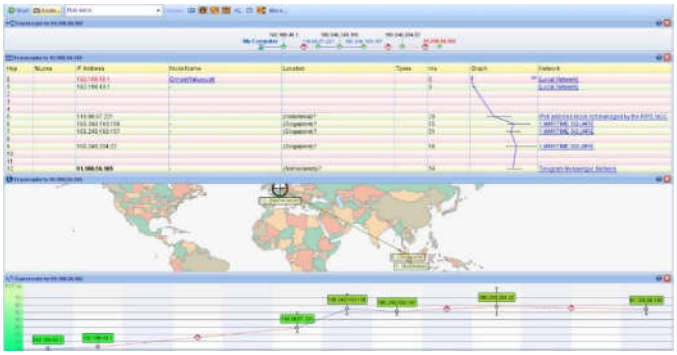
a

b

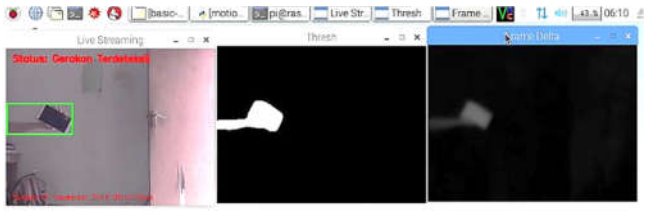
Gbr 2. (a) Kondisi saat mendapat sinyal *low*; (b) Kondisi normal dan saat mendapat sinyal *high*.



Gbr 3. *Screenshot* hasil *filter* ip.src == 192.168.137.5.



Gbr 4. *Screenshot* hasil *trace-route* dari IP 91.108.56.165.



Gbr 5. Contoh dari *threshold* dan *frame delta* pada sistem *surveillance.*

Tracert atau *trace-route* adalah proses penjejakan *rute traffic* dari *source* ke *destination* dengan memanfaatkan gema/pantulan atau echo paket *Internet Control Message Protocol* (ICMP) dan nilai *Time-To-Live* (TTL) dari paket yang dikirim menuju ke *destination*. *Tracer*t akan menghasilkan *list router*, jumlah hop, dan lama waktu yang dibutuhkan untuk mencapai destinasi. Sedangkan *who-is* merupakan protokol atau langkah untuk mengetahui info dari sebuah IP *address* atau domain yang dikelola oleh registrar.

Pada Gbr 4. dapat dilihat bahwa host 91.108.56.165 (Telegram) telah berhasil ditemukan, dan membutuhkan 12 hops untuk mencapai server Telegram_Mesenger_Network. Host tersebut merespon HTTP request pada port 80 (Telegram menggunakan server nginx/0.3.33 yang merespon dalam waktu 117ms) nilai TTL dari paket yang diterima dari server tersebut adalah 53.

C. Pengujian Sistem Surveillance

Program *surveillance* dijalankan dan file konfigurasi dipanggil dengan perintah "sudo python anu.py --conf conf.json", program akan memasuki proses inisialisasi dan OpenCV dalam tahap *pre-processing* dan inisialisasi *average background.* Proses inisialisasi *average background* adalah proses dimana program *surveillance* menghitung nilai *mean* (rata-rata) dari *previous frame* dan *current frame*

Berdasarkan nilai *mean* atau rata-rata dari *previous frame* dan *current frame*, diperoleh *average background model*. Setelah itu, *average background model* dikurangi dengan *current frame* sehingga diperoleh nilai selisih *frame* yang untuk selanjutnya disebut dengan *delta frame*. Kemudian, nilai *delta frame* tersebut dibandingkan dengan nilai *delta threshold.*

Delta *threshold* adalah nilai minimum atau nilai ambang-batas bawah dari Delta untuk pixel yang akan dideteksi sebagai gerakan, jika nilai *delta frame* lebih besar dari *threshold*, maka pixel pada tersebut akan diberi warna putih, dan dianggap sebagai *Foreground*. Dan sebaliknya, apabila nilai *delta frame* lebih kecil dari delta *threshold*, maka pixel akan diubah menjadi hitam dan dianggap sebagai *background* oleh sistem. Semakin kecil nilai dari delta *threshold* maka akan semakin sensitif sistem dalam mendeteksi gerakan. Nilai *delta threshold* untuk penelitian ini adalah 5. Angka ini diperoleh dari observasi (menyesuaikan kondisi cahaya dan *background*). Jika nilai *delta frame* lebih besar dari nilai *delta threshold*, maka gerakan yang terdeteksi akan diberi *contour* (garis disekitar gerakan). Namun jika nilainya lebih kecil maka sistem akan kembali menghitung *delta.frame*

Kemudian langkah selanjutnya adalah *contour detection* atau *edge detection*, langkah ini di tunjukkan oleh Gbr 5. Proses ini berfungsi sebagai penanda, bahwa adanya gerakan pada *frame* yang melebihi *minimum area*. Pada penelitian ini, tanda tersebut berupa garis berwarna hijau di sekitar gerakan.

Minimum area merupakan ukuran minimum (dalam satuan piksel) dari suatu *frame* untuk area yang akan ditandai sebagai gerakan atau tidak. Semakin kecil nilai *minimum area*, maka akan semakin sensitif pendeteksian atau semakin banyak area yang akan ditandai sebagai gerakan. Sedangkan semakin besar nilai *minimum area*, maka area yang akan ditandai sebagai gerakan akan semakin besar. Penentuan nilai *minimum area* disesuaikan dengan kebutuhan sistem. Pada sistem digunakan pada penelitian memiliki nilai *minimum area* sebesar 5000 pixel.

Jika *contour* lebih besar dari *minimum area*, maka status sistem akan berubah menjadi "Ada Gerakan!" dan sistem akan melakukan *capture* pada *current frame*. Disaat yang bersamaan sistem akan mengirim notifikasi ke Telegram *user* dan mengunggah hasil *capture* ke Dropbox.

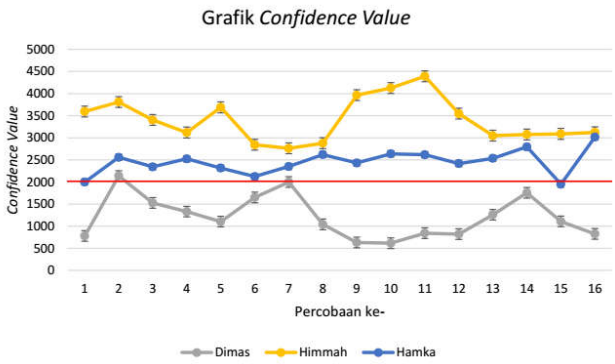
D. Pengujian Proses Training

Terdapat dua tahapan dalam pengujian ini, tahap pertama adalah proses registrasi wajah, dan tahap kedua adalah *training* sistem pengenalan wajah. Tahap registrasi wajah berfungsi untuk menghasilkan *training dataset* dengan gambar wajah yang diizinkan untuk membuka kunci pintu atau biasa disebut *positive dataset,* dan menyediakan contoh wajah-wajah yang tidak diizinkan untuk membuka kunci pintu atau biasa disebut *negative dataset.*

Saat *push button* ditekan, program akan melakukan *capture* menggunakan Pi-NoIR *Camera Module* dan dilanjutkan dengan konversi hasil *capture* menjadi citra *grayscale*. Selanjutnya program menjalankan *face detection* menggunakan metode *haar-cascades classifiers.* Bila ditemukan wajah, program akan melakukan *crop* disekitar wajah dan merubah ukuran wajah atau *resize* menjadi 92x112 pixel, kemudian menyimpan hasil *resize* kedalam direktori "*training/positive*". Peneliti menggunakan *database* yang berisi 100 gambar, dan masing-masing dari sepuluh orang yang berbeda untuk dimasukkan kedalam *negative training images* atau *negative dataset.*

Program *training* dimulai dari membaca *database positive dataset* dan *negative dataset.* Setelah semua gambar yang akan di-*training* sudah terbaca, akan dibuat model *Eigenfaces* berdasarkan *training dataset* tersebut. Ketika proses *training*

.



Gbr 6. Grafik *confidence value.*

model selesai, akan muncul pesan "[success] Training selesai.", hasil *training* disimpan menggunakan sintaks "model.save(config.training.xml)" ke dalam *file* berformat xml dengan nama "training.xml".

E. Pengujian Pengenalan Frontal Faces (Uji Klasifikasi)

Pada penelitian ini dilakukan pengujian sebanyak 48 kali pengujian pengenalan *frontal faces* dengan 1 buah wajah yang sudah dikenali oleh sistem dan 2 buah wajah yang belum dikenal oleh sistem. Pengujian dilakukan dalam 4 kondisi, yaitu pagi (06.00-07.00), siang (12.00-13.00), malam (18.00-19.00) dengan penerangan yang berasal dari lampu led 8 watt, dan gelap total (18.00-19.00) tanpa penerangan dari lampu, namun menggunakan bantuan led infra-merah untuk membantu teknologi *night vision*.

Dari 12 kali pengujian pagi hari (1050-4150 Lux), terdapat dua kali error. Masing-masing satu kali *false negative error*, dan satu kali *false positive error*. *False negative error* merupakan kondisi dimana sistem memberi keputusan bahwa *user* dengan wajah positif (teregistrasi) tidak bisa membuka kunci. Sedangkan *false positive error* adalah kondisi dimana sistem mengijinkan wajah tamu (yang tidak teregistrasi) bisa membuka kunci. *False negative error* terjadi karena nilai *confidence value* dari wajah *user* (Dimas) melebihi *threshold* yang telah ditetapkan. Sedangkan *false positive error* terjadi saat wajah tamu dengan nama Hamka memiliki nilai *confidence value* dibawah *threshold*. Sedangkan pengujian pada siang hari (5600-86400 Lux) dan malam hari (15-78 Lux), tidak terdapat *error* sama sekali.

Dari 12 kali pengujian pada kondisi gelap dengan tingkat pencahayaan 0 Lux. Terdapat satu buah *false positive error*. *False positive error* tersebut terjadi saat wajah tamu dengan nama Hamka pada pengujian ke-3 memiliki nilai *confidence value* dibawah *threshold* yang berarti wajah tersebut diizinkan untuk membuka pintu

Dapat dilihat pada Gbr 6, terdapat grafik yang menunjukkan garis nilai *confidence value* dari ketiga wajah, dan masing-masing wajah melakukan 16 kali percobaan. *User* Dimas yang wajahnya telah diregistrasi, rata-rata berada di bawah ambang batas, atau *threshold*. *User* Dimas hanya satu kali melewati *threshold*, yaitu pada percobaan ke 2. Sedangkan wajah Hamka dua kali berada diambang batas atau *threshold*, yaitu pada percobaan ke 1 dan ke 15. Sedangkan wajah Himmah tetap berada diatas ambang batas. Berdasarkan perhitungan 5, tingkat akurasi dari uji pengenalan wajah adalah sebesar 93,75%. Dengan nilai *error* sebesar 6,25%.

$$\frac{48-3}{48} \times 100 = 93{,}75\% \qquad (5)$$

V. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Program penguji sistem pengunci pintu ketika dijalankan akan memberikan sinyal *low* atau 0 pada GPIO.22 untuk menyalakan *solenoid*, kemudian jeda selama 0,2 detik, lalu memberikan sinyal *high* atau 1 pada GPIO.22 untuk mematikan *solenoid*. Arus yang mengalir pada kumparan yang terdapat di *solenoid* tersebut menghasilkan gaya elektromagnet yang akan menarik inti besi beserta tuas pengunci ke dalam dan pintu dapat dibuka.
2. Membutuhkan 12 hops untuk mencapai *host* 91.108.56.165 (Telegram_Mesenger_Network) yang berlokasi di Belanda. *Host* tersebut merespon HTTP *request* pada *port* 80 dalam waktu 117ms, nilai TTL dari paket yang diterima dari *server* tersebut adalah 53. Rute ini menawarkan throughput yang baik, dengan Hop yang merespon rata-rata dalam waktu 36ms. Dan DNS *lookup* dan *who-is* diselesaikan hampir dalam sekejap (kurang dari 2ms).
3. Sistem *surveillance* akan mendeteksi gerakan Jika *contour* lebih besar dari *minimum area*. Kemudian sistem akan melakukan *capture* pada *current frame*. Disaat yang bersamaan sistem akan mengirim notifikasi ke Telegram *user* dan mengunggah hasil *capture* ke Dropbox
4. *User* Dimas hanya satu kali melewati *threshold*, yaitu pada percobaan ke 2. Sedangkan wajah Hamka dua kali berada diambang batas atau *threshold*, yaitu pada percobaan ke 1 dan ke 15. Sedangkan wajah Himmah tetap berada diatas ambang batas. Tingkat akurasi dari uji pengenalan wajah adalah sebesar 93,75%. Dengan nilai *error* sebesar 6,25%.

REFERENSI

[1] Eleyan, A., & Demirel, H. 2006. PCA and LDA Based *Face recognition* Using Feedforward Neural Network Classifier. Proc. of Multimedia Content Represenatation. , Istanbul, Turkey: Springer-Verlag.

[2] Janssen, Cory. Internet of Things: IoT. Diakses oleh id.wikipedia.org dari situs techopedia pada 9 November 2013.

[3] Cohen, N. Gattuso, J., & MacLennan-Brown, K. 2007. CCTV Operational Requirements Manual. UK: Home Office Scientific Development Branch

[4] Gandhe. S.T., Talele, K.T., & Keskar, A.G.. 2007. *Intelligent Face recognition Techniques: A Comparative Study*. GVIP Journal. vol. 7

[5] Suranata, A. 2015. Pengolahan Citra Untuk Deteksi Wajah Menggunakan Python Dalam 25 Baris Kode. Diakses dari situs tutorkeren.com pada 24 Desember 2017.