Full Paper

Design of Security System with Face Recognition Using Arduino and OpenCv

Dwi Ely Kurniawan[1], Aldo Abellto[2]
[1,2]Department of Informatics Engineering, Politeknik Negeri Batam, Indonesia

Article Info

Article history:

Received 22 August 2021
Revised 12 September, 2021
Accepted 18 October, 2021

Keywords:

Security System
Face Recognition
Arduino
OpenCv
IOT

ABSTRACT

Security system is the most developed technology today, various technologies such as from CCTV cameras, access door, encrypt password to use biometric technology. Biometric technology is a self-recognition technology using body parts and human behavior that has a certain uniqueness. One example of application of Biometric technology is face recognition or face recognition system. Security mengunkanan Biometric technology is very difficult to be broken in because every human has its own characteristics and uniqueness. Security system design using face recognition is using webcam camera, Laptop, Arduino device and OpenCv library. The webcam camera will be attached to the laptop, then the webcam receives an image input and proceeds to use the OpenCv library then the signal is sent to the Arduino device, if access is received the blue LED will light up if not the red LED will light up. How users can access a data using face recognition security system. The sample test takes the time delay required to receive the response from the camera, how much data is suitable but does not receive the appropriate response, the effect of distance to the camera, and the effect of the angle of the face. Testing the sample only until the LED lights menyalanya accordance with the access provided.

Corresponding Author:

Dwi Ely Kurniawan,
dwialikhs@polibatam.ac.id

1. INTRODUCTION

The biometric system is a self-recognition Security System technology by using body parts or human behavior that have unique characteristics. One part of the biometric system is face recognition. Facial recognition systems are widely used for personal identification such as the use of attendance machines, door access control, and others [16].

Face recognition is one solution that can be used to secure a system. Face recognition requires digital image processing or image processing which will later process or process data with input in the form of images (images) and transformed into other images as output with certain techniques. OpenCV is the library most often used to perform image processing [13]. This face recognition uses the Principal Component Analysis (PCA) method which is used to reduce dimensions to produce orthogonal-based vectors called eigenfaces[2]. Principal Component Analysis (PCA) aims to take facial features which will later be processed and then used to recognize a person's face.

At this time, the use of devices to secure a system does not have to use only one device. The combination of several devices can help make the system that is made more secure and easier to control. Not only merging devices can have an effect, combining several self-recognition methods such as combining facial recognition with fingerprint so that the system made is more secure. This system is also integrated with an Arduino microcontroller which is used to receive a response from a facial recognition system application, then the response will be used to turn on the LED as a security system prototype.

The result of this research is to be able to integrate the face recognition algorithm with Arduino and OpenCV and analyze the required delay, light, and face position. It is hoped that it can help the development

of the Security System to be more secure from external threats, by using facial recognition technology to identify users who will access data.

2. RESEARCH METHOD

The facial recognition system that is integrated with the Arduino microcontroller has 4 main hardware components that are connected.

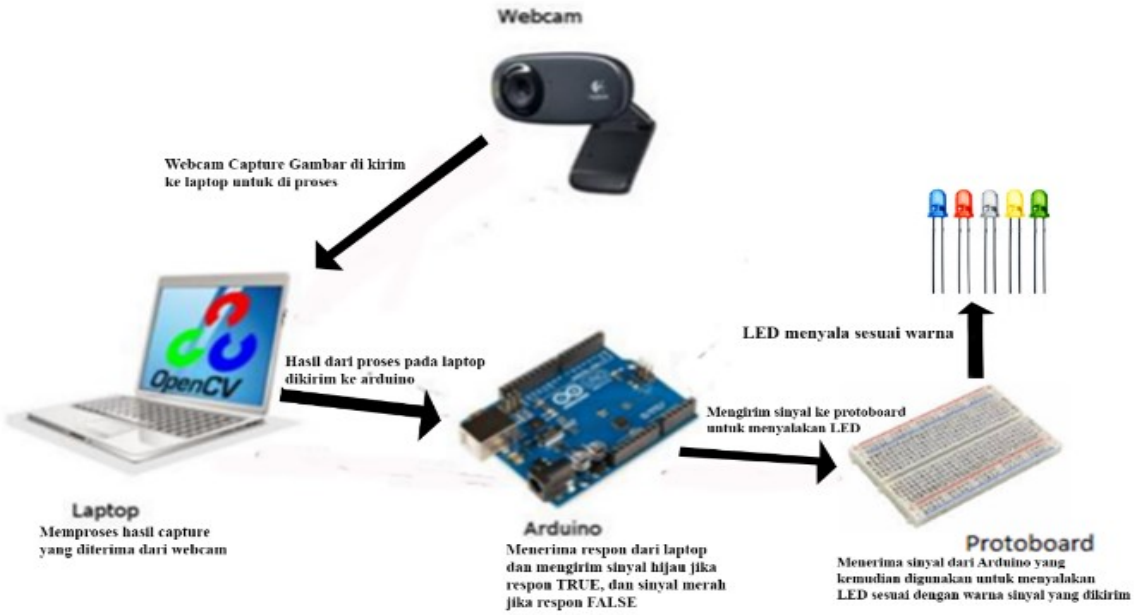


Figure 4. Overview of the main hardware components and the security system prototype process flow

The process of the security system using face recognition is divided into 2 parts of the process, namely, the image recording process and the image recognition process. The image recording process is the process of taking sample data that is saved to the face database, later it will be used as a comparison in the recognition process, while the recognition process is the process of recognizing images captured by the webcam and matched into the face database.

The first is the process of recording facial data which will be used as sample data for face recognition:

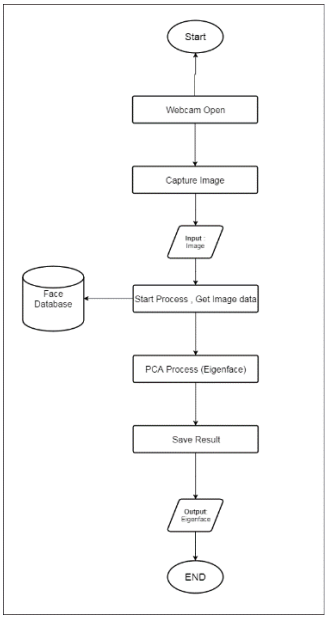


Figure 5. Flowchart of face recording process.

• Webcam is activated to display the image to be captured in the application.
• Face capture is done in real-time by the webcam, if a face is detected in the image, the system will capture the image.
• Perform initial processes such as normalizing image size, changing RGB image color to grayscale, performing histogram equalization to improve image quality to obtain clear image information.
• Save the captured face data in the form of *. jpg.
• Perform the PCA process to get the most important part using the eigenface method so that the eigenvector and eigenvalue of the image are obtained.
• Storing the results of the PCA process in a formal XML form, which contains image information when performing eigenface calculations.
• The data that has been saved will be used as a comparison value in the face recognition process.

The second performs facial recognition, by analyzing the sample data that has been previously stored:

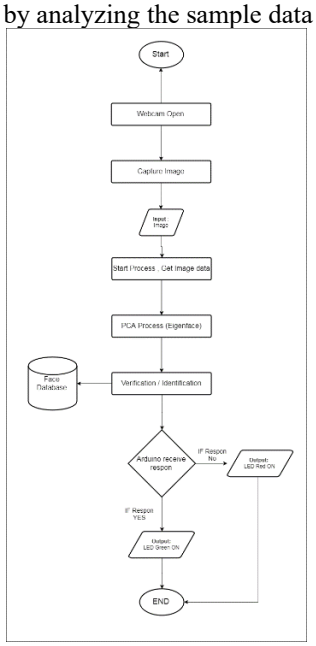


Figure 6. Flowchart of the face recognition process.

• Webcam is activated to display the image to be captured in the application.
• Face capture is done in real-time by the webcam, if a face is detected in the image, the system will capture the image.
• Perform initial processes such as normalizing image size, changing RGB image color to grayscale, performing histogram equalization to improve image quality to obtain clear image information.
• Perform the PCA process to get the most important part using the eigenface method so that the eigenvector and eigenvalue of the image are obtained.
• Then perform face recognition by calculating the distance between facial features, the distance obtained is the smallest for identification. The more sample data, the higher the level of accuracy in facial identification.
• The results of the identification are sent to the Arduino microcontroller to control the installed LEDs,
• The LED will light up according to the rules that have been determined, if access is accepted the blue LED will light up, but if it is not accepted then the red LED will light up.

Designing the LED circuit path using a protoboard that is connected to Arduino, Arduino will later control this protoboard according to the circuit path that has been determined, the following is the LED circuit path using Arduino:

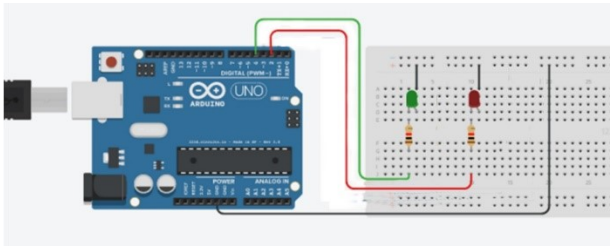


Figure 7. Overview of the LED circuit installed on the Arduino microcontroller.

The above scheme uses Arduino Uno R3 and Protoboard, besides that there are several more components such as:
1. 2 Pcs 5mm LEDs or 3mm LEDs (Red and Green Colors)
2. 2 Pieces of 100Ω Ohm Resistors
3. 3 Pcs Jumper Cables
4. 1 Piece Protoboard
5. 1 Arduino

All tools and materials are connected to form a complete circuit.
1. Arduino GND Pin (Black) to GND/CATHODE LED0, 1, 2, 3, and 4 (Black)
2. Pin 4 (Green) to Resistor
3. Pin 2 (Red Color) to Resistor
4. The other side of all the resistors is connected to the positive LED (1 LED = 1 resistor and so on)

3. RESULTS AND ANALYSIS

Testing the security system using face recognition is carried out by taking samples of 10 users, each of which has their faces recorded 10 times so as to produce 100 face data stored in the database.

Figure 8. Overview of a sample collection of user images.

This facial position test uses 5 users each doing 20 trials consisting of 10 times the normal position, 5 times the abnormal position that is not trained, 5 times the abnormal position that is trained.

Table 1. Result of experiment

No	Trials	Normal (50 times)	Abnormal (25 times)	Abnormal Training (25 times)
1	Success	43 Kali	18 Kali	21 Kali
2	Error -Unknown	0 Kali	2 Kali	0 Kali
3	Error - User	7 Kali	5 Kali	4 Kali
Persentase **Success**		86%	72%	84%
Persentase **Error**		14%	28%	16%

The facial photo test uses a photo that has been printed containing the face of the user. The photos used are divided into 2 types of photos, photos containing one person and photos containing 2 people. Each type of photo will be tested 10 times.

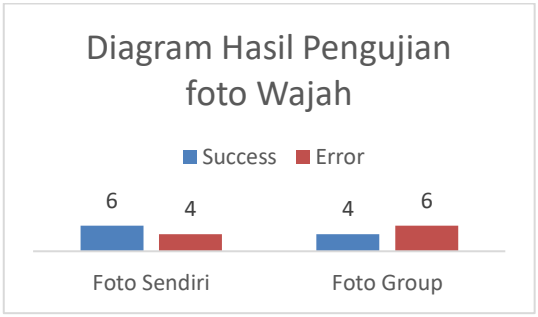
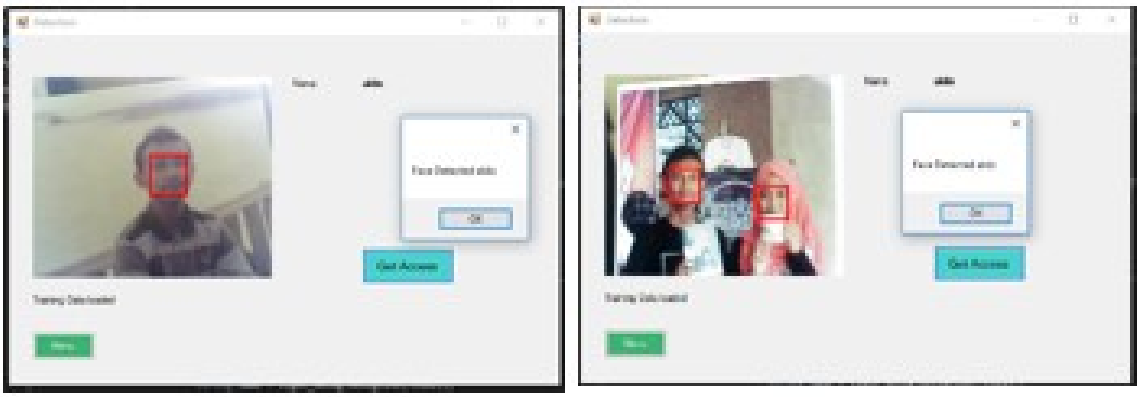


Figure 8. Graphical image of the results of the facial photo test.

In the use of photos alone, the success percentage is 60%, for group photos or two people the success is 40% depending on the position of the user in the photo, the system reads face images sequentially from the right, and only processes 1 face.



Testing the effect of light is carried out with 2 light situations, dark light situations and bright light situations. Each light situation was tested 20 times.

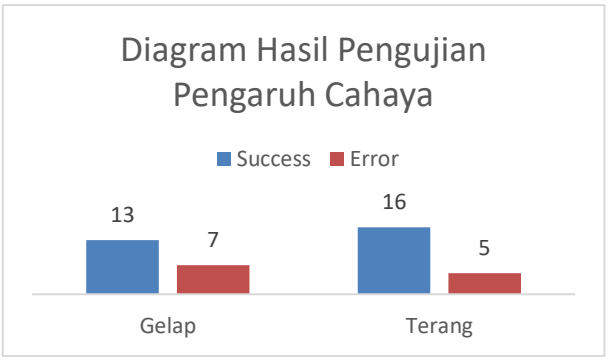


Figure 11. Graphical image of the results of testing the effect of light.

In bright light or cupuk the success rate of the experiment is higher than in dark light, because the light received by the camera is sufficient so that the image capture results are clearer and the user recognition process is more accurate.

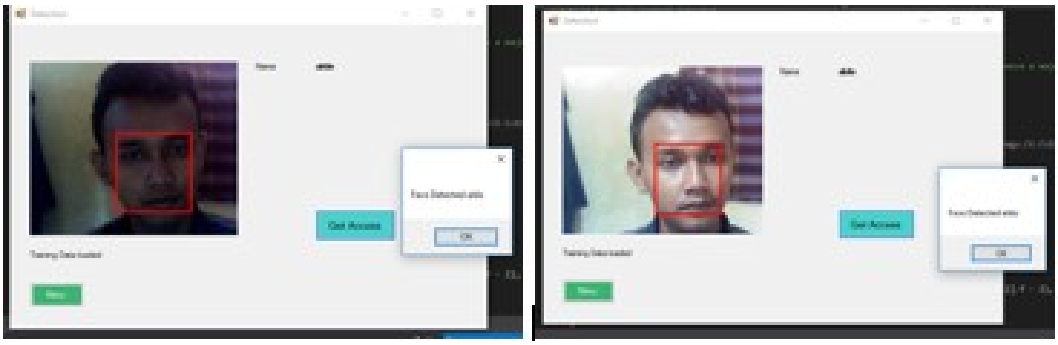


Figure 12. Picture of test results.

Testing the time delay required for the system to send and receive responses from the application to Arduino, the test was carried out 20 times by one user to determine the average delay needed.

Table 2. Result of Detection Status

No	Delay (ms)	*Detection Status*
1	165	Success
2	135	Success
3	140	Error
4	144	Success
5	150	Success
6	148	Success
7	140	Error
8	136	Success
9	133	Success
10	125	Success
11	130	Success
12	139	Success
13	141	Success
14	149	Error
15	155	Success
16	165	Success
17	146	Success
18	144	Success
19	135	Success
20	165	Success
Average	143.2 ms	

Face recognition applications are divided into 2 basic systems, namely facial recognition systems and data record systems. To test the two systems there are 4 processes:

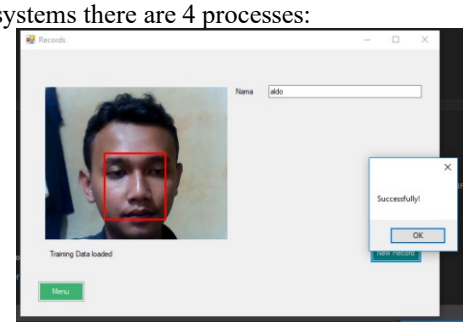


Figure 13. Image of data records.

Recording the face of the user to be registered, in this study the data stored was 10 users with 10 photos for each user.

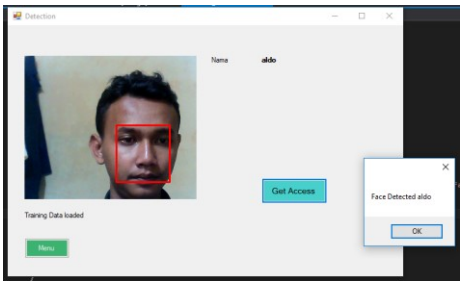


Figure 14. Image of face detection.

Conduct an experiment to detect whether the user's face is in the database.

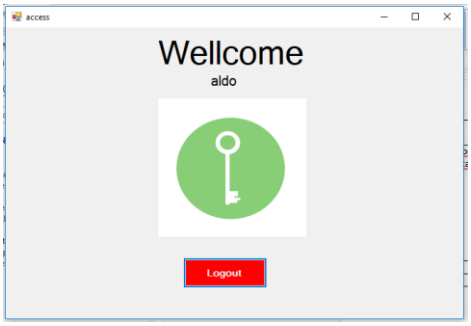


Figure 15. Image if the face is registered in the database.

If it is successful and the face to be detected is in accordance with the data contained in the database, a view like the image will appear, and the system will send a signal to the Arduino microcontroller to turn on the prototype LED. If the signal sent is TRUE then the green LED will light up, if FALSE then the red LED will light up.

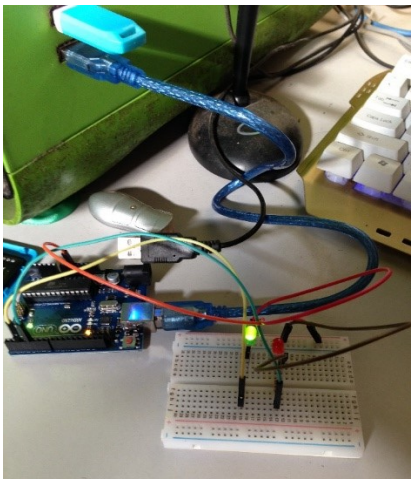


Figure 16. The green LED lights up when the response is that the user is registered in the system database.

The overall system testing was carried out 30 times and used 1 user with normal conditions, following the overall test results.

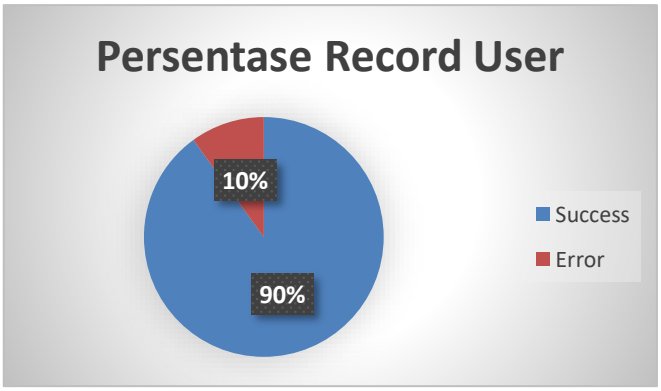


Figure 17. Picture of the success percentage of Record User.

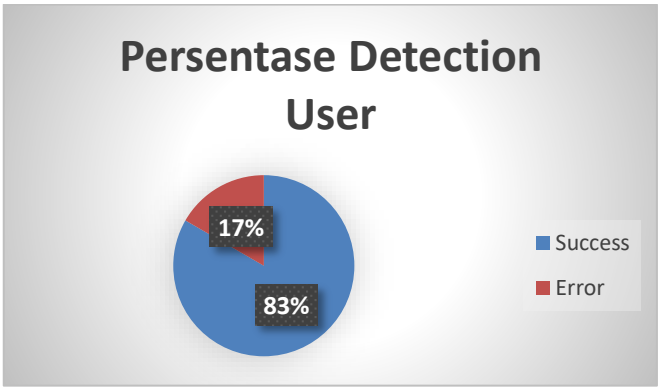


Figure 18. Image of the percentage of successful Detection User.

4. CONCLUSION

After carrying out the stages of implementing security system testing using face recognition, the conclusions of this study were obtained as follows:

1) The PCA method uses the Haar Feature algorithm to detect facial areas by reading certain points on the face.
2) The eigenface algorithm can perform face recognition by training the existing data and then looking for the eigenvalues used to see how well the training data matches the detected faces.
3) Face detection process can be done in real-time
4) From the testing of 3 types of facial positions, the system is more accurate when the user's face is in a normal position with a success percentage of 86% for 50 trials.
5) Abnormal position of the face, such as wearing a hat, if training has been carried out, it will get a fairly high percentage of success up to 84%
6) The user detection accuracy rate in this study was 83.3%.
7) The delay required to send and receive a response from the application to the Arduino averaged 143.2 ms.
8) The more sample data stored in the database per person, the more accurate this system will be.
9) Arduino can be combined with desktop applications
10) The error rate of detecting faces that have sibling relationships is quite high because they have almost similar facial features so that when doing training the eigenvalues are close to.

REFERENCES

[1] A. Aditya Shankar dan P.R.K.Sastry dan A. L.Vishnu Ram dan A.Vamsidhar, 2015, Finger Print Based Door Locking System, IJECS Volume 4 Issue 3 March, 2015 Page No.10810-10814.
[2] Bayu, Setya dan Hendriawan, Achmad dan Susetyoko, Ronny, 2009, Penerapan Face Recognition Dengan Metode Eigenface Dalam Intelligent Home Security, Eepis Final Project.
[3] [Cascade Classification, 2014, Accessed December 19, 2017.

[4] https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
[5] Ecadio, 2015, Accessed September 28, 2017. http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3
[6] Metode Algoritma, 2013, Accessed Oktober 2, 2017. http://www.warriornux.com/membuat-lampu-led-berjalan-menggunakan-arduino-uno/
[7] Handika, Yola Tri, 2014, Implementasi Metode Filter Gabor Dan Backpropagation Neural Network Pada Sistem Pengenalan Wajah Sebagai Personal Authentication
[8] JANNAH, Miftachul, 2012, Sistem Pengenalan Wajah 2-D Menggunakan Jaringan Syaraf Tiruan Algoritma Backpropagation Dan Metode Eigenface Principal Component Analysis, Skripsi, Universitas Pendidikan Indonesia, Bandung.
[9] Kurniawan, Dwi Ely, 2014, Sistem Identifikasi Biometrika Wajah Menggunakan Metode Gabor KPCA dan Mahalanobis Distance, JSINBIS (Jurnal Sistem Informasi Bisnis), Universitas Diponegoro, Semarang.
[10] Isbadi, Urifan dan Risanuri , Hidayat dan Indah, Soesanti, 2010, Pengenalan wajah dengan metode eigenface, Jurnal Penelitian Teknik Elekro Vol 3 No.4 , FT Universitas Gajah Mada, Yogyakarta.
[11] Maarik, Ardhi dan Shaztika , Sarah Shanaz dan Radhi, Muhammad S dan Juliani, Adek A. P, 2015, Object Trcaking menggunakan OpenCV dan Arduino, Laporan Tugas Akhir PPCD, Bogor Agricultural University, Bogor.
[12] Maulana, Hari Rahmat, 2016, Pembacaan Plat Nomor Kendaraan Otomatis Menggunakan Kamera Berbasis OpenCv Dengan Metode Neural Network, Tugas Akhir, Politeknik Negri Batam, Batam.
[13] [13] Metode Algoritma, 2013, Accessed September 28, 2017. http://www.metode-algoritma.com/2013/06/algoritma-eigenface-pendeteksian.html
[14] N. Putri dan G. Wikaria dan B. Widodo, Sistem Kontrol Akses Berbasis Real Time Face Recognition Dan Gender Information, Comtech, Vol.6 No.2 Juni 2015: 198-207.
[15] Priawadi, 2012, Accessed September 24, 2017. http://www.priawadi.com/2012/09/opencv.html
[16] Ramadani. Deni,2011, Aplikasi Perbandingan Algoritma Metode Fisherface Dengan Metode Eigenface Pada Sistem Pengenalan Pola Wajah, Skripsi, Universitas Komputer Indonesia, Bandung.
[17] Sitorus, Givson Apsyan, 2016, Deteksi Wilayah Wajah Dan Pengenalan Wajah Menggunakan Webcam Berbasis OpenCv, Tugas Akhir, Politeknik Negri Batam, Batam.
[18] S.V. Viraktamath, 2013, Face Detection and tracking using OpenCv, Thesij CNCE-0103540102.
[19] Wariornux, 2016, Accessed September 29, 2017. http://www.metode-algoritma.com/2013/06/metode-eigenface-pengenalan-citra-wajah.html.