

Deteksi dan Pengenalan Wajah sebagai Pendukung Keamanan Menggunakan Algoritme *Haar-Classifler* dan *Eigenface* Berbasis Raspberry Pi

Hernanda Agung Saputra¹, Fitri Utamingrum², Wijaya Kurniawan³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹agunghernanda17@gmail.com, ²f3_ningrum@ub.ac.id, ³wjaykurnia@ub.ac.id

Abstrak

Salah satu hal yang tidak terlepas dari kemajuan teknologi adalah keamanan. Jika hanya mengandalkan sistem keamanan menggunakan tenaga manusia, hal tersebut juga tidak begitu efektif karena manusia juga mempunyai rasa lelah. Maka dari itu diciptakan sistem pendukung keamanan seperti *barcode*, *rfid card*, *password*, *PIN* dan sebagainya. Namun penggunaan media keamanan tersebut mempunyai beberapa kelemahan yaitu dapat hilang, rusak, dan dicuri atau disalahgunakan oleh orang yang tidak bertanggung jawab. Salah satu alternatif yang dapat dilakukan yakni memanfaatkan wajah sebagai *data* keamanan. Pada penelitian ini sistem yang dibuat menggunakan *Raspberry Pi 3* yang diintegrasikan dengan *webcam Logitech C525* sebagai *input*, serta mikrokontroler *Arduino Uno* sebagai pemroses *sensor* ultrasonik dan cahaya. Untuk *buzzer*, *LCD*, dan modul *SIM800L* digunakan sebagai *output* dari sistem untuk memberikan notifikasi berupa *alarm*, *visual* dan *SMS*. Sistem ini menggunakan *Haar-Classifler* untuk mendeteksi objek wajah pada citra yang ditangkap oleh *webcam*. Selanjutnya digunakan metode *Eigenface* untuk menghasilkan bobot citra wajah. Setelah bobot citra wajah didapat, dicari selisih terkecil bobot citra wajah baru dengan bobot citra wajah pada *database*, yang dimana hasil tersebut menentukan bagaimana *output* dari sistem nantinya. Dari hasil pengujian akurasi deteksi wajah didapatkan akurasi deteksi wajah terbaik yaitu pada jarak 40 cm sebesar 100%. Hasil akurasi pengujian pengenalan wajah pada jarak 40 cm secara keseluruhan yaitu sebesar 75%. Dari pengujian integrasi sistem *software* dengan *hardware* didapatkan presentase *error* sebesar 0%. Waktu rata-rata komputasi dalam mengenali wajah adalah sebesar 0.11536 detik.

Kata kunci: *raspberry pi 3, arduino-uno, HaarClassifler, Eigenface, Euclidian Distance, face-detection, face-recognition, security*

Abstract

One of the things that is inseparable from the progress of technology is security. If we only rely on a security system using human power, it is also not so effective because people also have a sense of tired. Therefore a security support system was created such as barcode, rfid card, PIN, password and etc. However, the use of media that has some security flaws namely can be lost, stolen or damaged, and abused by people who are not responsible. One of the alternatives that can be performed i.e. utilize face as data security. On the research of this system are made using Raspberries Pi 3 were integrated with the Logitech webcam C525 as input, as well as the mikrokontroler Arduino Uno as ultrasonic and light sensor processing. For LCD, buzzer, and module SIM800L is used as the output of the system to provide notification in the form of a alarm, visual text, and SMS. This system uses Haar-Classifler to detect face objects in the image captured by the webcam. Next, Eigenface method is used to get weight of face image. After weight of face image obtained, search the smallest difference in weight of face image of new faces with the image of the face on the database where the results determine how the output from the system. From the results of testing the accuracy of face detection, best accuracy is obtained at a distance of 40 cm with 100% accuracy. Overall accuracy of testing the accuracy of face recognition at a distance of 40 cm is 75%. From system integration testing software with hardware obtained percentage error of 0%. The average time of computation in recognizing a face is 0.11536 seconds.

Keywords: *raspberry pi 3, arduino-uno, HaarClassifler, Eigenface, Euclidian Distance, face-detection, face-recognition, security*

1. PENDAHULUAN

Setiap waktu teknologi berkembang sangat pesat. Hal ini bisa ditemukan pada penggunaan teknologi yang diimplementasikan di kehidupan sehari-hari. Dikarenakan hal tersebut banyak orang yang bergantung pada teknologi yang ada. Salah satu hal yang tidak terlepas dari kemajuan teknologi adalah keamanan.

Setiap sistem keamanan yang ada pasti mempunyai kelemahan masing-masing. Jika hanya mengandalkan sistem keamanan menggunakan tenaga manusia, terkadang hal tersebut juga tidak begitu efektif karena manusia sejatinya juga mempunyai rasa lelah. Dikarenakan hal tersebut maka diciptakan sistem pendukung keamanan dengan menggunakan media keamanan seperti *barcode, rfid card, password, PIN* dan sebagainya. Namun penggunaan media keamanan tersebut sebenarnya mempunyai beberapa kelemahan yaitu dapat hilang, rusak, dan dicuri atau disalahgunakan oleh orang lain yang tidak bertanggung jawab.

Wajah manusia mempunyai banyak informasi. Selain dapat memperlihatkan suasana hati, niat, dan perhatian, wajah juga dapat berfungsi untuk mengidentifikasi seseorang. Wajah adalah kunci paling khas dan banyak digunakan untuk identitas seseorang (Bruce & Young, 2018). Biometrik merupakan ciri khas yang dapat digunakan pada suatu sistem keamanan yakni pengenalan wajah sebagai identitas *data* (Budiawan & Andriana, 2014). Salah satu teknik identifikasi yang diterapkan pada teknologi biometrik yaitu menggunakan wajah sebagai parameter utama merupakan pengertian dari pengenalan wajah (Face recognition) (Susanto, et al., 2017).

Metode yang digunakan dalam untuk pengenalan wajah sebenarnya cukup banyak. Salah satu metode pengenalan wajah yaitu *Eigenface*. Dalam metode *Eigenface* tidak semua *piksel* dari citra wajah digunakan, hanya diambil ciri dari citra wajah tersebut saja. Sehingga komputasi menjadi lebih sederhana dan waktu komputasi yang diperlukan untuk mengenali citra wajah tidak terlalu lama (Wasista, et al., 2011). Dengan demikian metode *Eigenface* cocok diterapkan pada perangkat *embedded* dengan spesifikasi yang terbatas (Yanto, et al., 2016). Cara kerja dari metode *Eigenface* yaitu dengan melakukan perhitungan

pada citra wajah yang ada pada *database* dan menghasilkan nilai *Eigenvalue* serta *Eigenvector* yang dimana kedua nilai tersebut digunakan untuk mencari dan menghasilkan nilai bobot pada setiap citra wajah yang ada pada *database*. Cara sistem untuk mengetahui apakah citra wajah baru yang terdeteksi, dikenali dan terdapat pada *database* yaitu dengan mencari selisih terkecil menggunakan *euclidian distance* (Turk & Pentland, 1991).

Berdasarkan permasalahan yang sudah dijelaskan diatas, maka perlu adanya sebuah sistem pendukung keamanan yang bisa mengatasi kelemahan-kelemahan yang sudah disebutkan sebelumnya dengan cara menggunakan pengenalan wajah sebagai *data* keamanan utama.

Penelitian sebelumnya yang dilakukan oleh (Budiawan & Andriana, 2014) mengangkat judul Pengujian Pengenalan Wajah Menggunakan Raspberry Pi. Penelitian ini berhasil dalam mengenali wajah menggunakan metode *Eigenface* dan menggunakan perangkat Raspberry Pi dengan persentase *equal error rate (EER)* yang kecil yaitu sebesar 13,333%. Pada penelitian ini, sistem tidak dilengkapi dengan perangkat yang dapat menunjang keamanan seperti buzzer dan perangkat keamanan yang lain sebagai tanda bunyi atau yang lainnya.

Penelitian sebelumnya yang berkaitan dengan pengenalan wajah menggunakan algoritme *Eigenface* dan *Haarcascade* oleh (Nicco & Fahrzy, 2014). Penelitian berfokus pada kondisi *real-time* dengan menggunakan komputer sebagai otak atau *CPU* utama dalam memproses *data*. Hasil akurasi dari penelitian ini cukup baik yakni sebesar 83,33% dengan menggunakan 150 citra wajah sebagai *data training* dan 150 sebagai *data uji* dengan menggunakan jarak sebagai parameter sebesar 30 cm untuk mengukur tingkat keakuratannya. Pada penelitian ini tidak menggunakan perangkat *embedded* sebagai otak atau *cpu* sebagai media pemrosesan *data* serta tidak disebutkan berapa nilai intensitas cahaya yang digunakan.

Penelitian lain yang menjadi dasar penelitian ini yaitu berkaitan dengan metode *Eigenface* yang dilakukan oleh (Yanto, et al., 2016). Pada penelitian tersebut dilakukan implementasi algoritme *Eigenface* untuk pengenalan wajah dengan menggunakan Intel Galileo sebagai sistem keamanan ruangan. Tingkat akurasi pada

sistem tersebut sudah cukup baik yaitu mencapai 90%. Dari semua penelitian tersebut tidak disebutkan berapa tingkat pencahayaan saat sistem mulai berjalan untuk melakukan pengujian pengenalan wajah yang dilakukan serta alat untuk menunjang keamanan yang digunakan masih menggunakan *buzzer*.

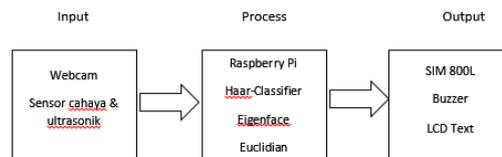
Hal tersebut yang membuat penelitian tersebut menjadi dasar bagi penulis untuk membangun sistem pengenalan wajah sebagai pendukung keamanan menggunakan metode *Eigenface* berbasis Raspberry Pi. Sistem bekerja saat *sensor* ultrasonik mendeteksi adanya objek pada jarak minimal yang sudah diatur sebelumnya, kemudian *webcam* yang terhubung ke perangkat Raspberry Pi akan mengambil citra pada saat itu juga. Citra yang didapatkan kemudian akan dideteksi keberadaan objek wajah dengan menggunakan algoritme *HaarCascade* dan hasil deteksi objek wajah tersebut akan *dicrop* dan disimpan sebagai gambar dengan ukuran 92 x 112 *pixels*. Selanjutnya dari gambar atau citra wajah baru yang didapat akan dicari bobot dengan menggunakan algoritme *Eigenface* yang dimana bobot tersebut akan dibandingkan dengan bobot setiap citra wajah yang ada pada *database* menggunakan algoritme *Euclidian Distance*.

Jika citra wajah baru yang dideteksi dikenali dan terdapat pada *database* maka sistem tidak akan mengeluarkan bunyi, LCD akan menampilkan mirip dengan salah satu citra *database* serta tidak mengirim sms peringatan. Namun apabila sistem tidak mengenali citra wajah baru tersebut maka alarm atau *buzzer* akan berbunyi, LCD tampil 'tidak mirip' dan sms peringatan akan dikirim sebagai indikasi adanya penyusup.

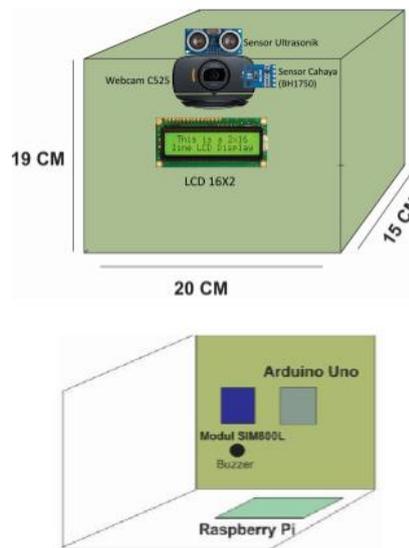
2. PERANCANGAN DAN IMPLEMENTASI

2.1 Gambaran Umum Sistem

Pada Gambar 1 merupakan blok diagram sistem. Sistem yang akan dibuat adalah sistem yang dapat mendeteksi dan mengenali objek wajah seseorang sesuai *data* yang ada pada *database*. Dimana data berupa gambar wajah akan diperoleh dari *webcam* USB sesuai perintah yang didapatkan pada saat *sensor* ultrasonik mendeteksi adanya objek pada jarak minimal yang sudah ditentukan. Selanjutnya gambar



Gambar 1. Blok Diagram Sistem



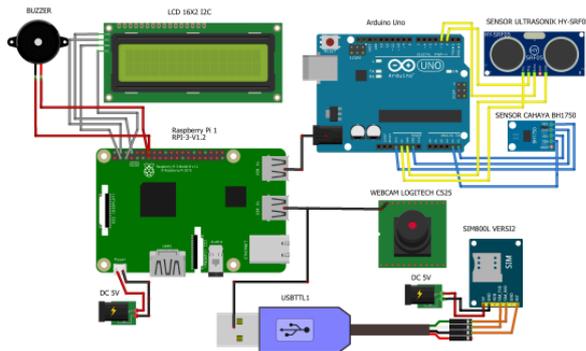
Gambar 2. Desain *prototype* alat

yang telah didapat dilakukan deteksi objek wajah dengan menggunakan algoritme *HaarClassifier*. Kemudian gambar objek wajah akan diolah oleh Raspberry Pi dengan menggunakan algoritme *Eigenface*. Hasil *data* yang sudah diolah oleh Raspberry pi akan berupa nilai bobot wajah. Nilai bobot wajah baru akan dibandingkan dengan bobot wajah yang ada pada *database* yang sudah diolah sebelumnya, menggunakan *Euclidian Distance*.

Apabila nilai bobot wajah baru ternyata cocok atau mendekati dengan salah satu bobot yang ada pada *database* maka *buzzer* tidak akan berbunyi, LCD menampilkan *teks* sesuai citra yang mirip serta sistem tidak akan mengirimkan SMS peringatan. Namun jika nilai bobot wajah baru tidak ada yang mendekati atau cocok pada nilai bobot wajah yang ada di *database*, maka *buzzer* akan berbunyi, LCD menampilkan *teks* tidak mirip dan sistem akan mengirim SMS peringatan ke *user*.

2.2 Perancangan Sistem

Pada Gambar 2 merupakan desain *prototype alat* yang akan dibuat. Desain ini dibuat untuk mengetahui posisi peletakan tiap komponen elektronik dan ukuran dari alat yang akan dibuat.

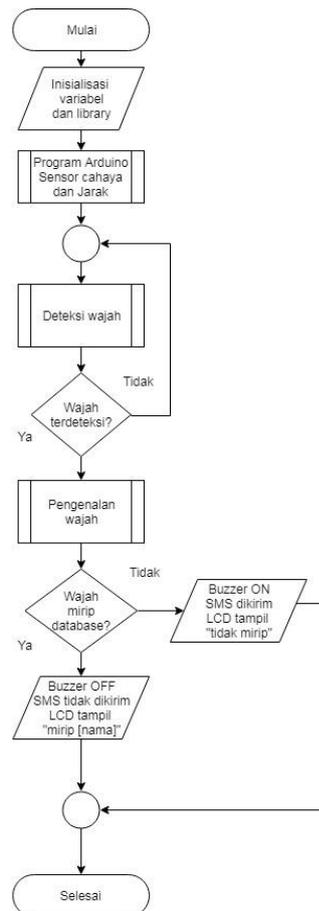


Gambar 3. Skematik Rangkaian Sistem

Tabel 1. Koneksi Pin Sistem

Pin Raspberry	Pin SIM800L	Pin USB TTL	Pin Buzzer	Pin LCD	Pin Camera Logitech C275	Pin Arduino	Pin Sensor Ultrasonik	Pin Sensor Cahaya
						5V	VCC	
						D2	TRIG	
						D3	ECHO	
						3,3V		VCC
						A4		SDA
						A5		SCL
GPIO 17				VCC				
5V	VCC			VCC				
GPIO 2				SDA				
GPIO 3				SCL				
		TXD	RX					
		RXD	TX					
USB		USB						
GND	GND	GND	GND	GND		GND	GND	GND

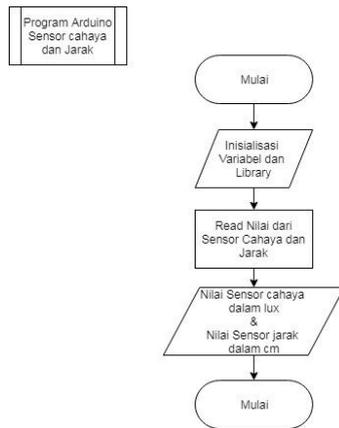
Berdasarkan Gambar 3, dapat dilihat bahwa sensor ultrasonik dan cahaya tersambung ke Arduino, sedangkan untuk catu daya dan komunikasi antara Arduino dengan Raspberry Pi dilakukan secara serial atau UART menggunakan slot USB. Webcam Logitech tersambung ke Raspberry Pi menggunakan slot USB. Untuk modul SIM800L tersambung ke pin USB TTL kemudian USB TTL tersebut tersambung ke slot USB Raspberry Pi. Komunikasi data yang digunakan oleh modul SIM800L disini adalah komunikasi serial atau UART. Untuk modul LCD I2C 16X2 dan buzzer tersambung ke Raspberry Pi dengan memanfaatkan pin GPIO yang terdapat pada Raspberry Pi. Dapat dilihat pula untuk sensor ultrasonik dan cahaya tersambung ke Arduino, kedua sensor tersebut digunakan mendeteksi jarak objek dengan webcam serta berapa nilai intensitas cahaya saat pengambilan gambar dilakukan oleh webcam Logitech yang tersambung pada Raspberry Pi. LCD 16X2 berfungsi untuk menampilkan nilai yang dibaca



Gambar 4. Diagram Alir Program Utama Sistem

dari sensor ultrasonik dan cahaya. Sedangkan buzzer dan modul SIM800L berfungsi sebagai indikasi bila ada objek wajah yang terdeteksi dan tidak dikenali oleh sistem. Selanjutnya pada Tabel 1. merupakan koneksi dari pin sistem yang dibuat.

Pada Gambar 4 merupakan proses utama dalam perancangan perangkat lunak. Dapat dijelaskan secara keseluruhan yaitu sistem ini mulai berjalan dengan menginisialisasi variabel dan library, kemudian masuk ke subprogram Arduino yaitu program sensor cahaya dan jarak, kemudian ke subprogram deteksi wajah. Jika wajah terdeteksi maka akan menuju ke proses selanjutnya yaitu subprogram pengenalan wajah untuk memproses citra wajah yang sudah didapatkan sebelumnya, namun apabila objek wajah masih belum ditemukan pada gambar, maka sistem akan mengambil gambar secara terus-menerus sampai mendapatkan atau mendeteksi objek wajah pada gambar yang diambil. Setelah melewati subprogram pengenalan wajah, maka output yang dihasilkan adalah apakah wajah mirip database atau tidak. Jika wajah mirip pada salah satu database maka

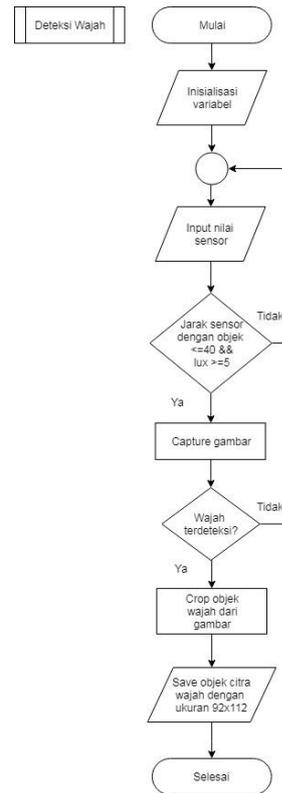


Gambar 5. Diagram Alir Program *Sensor Cahaya dan Jarak* Arduino

output *Buzzer OFF*, SMS tidak dikirim serta LCD tampil mirip [nama], namun apabila wajah tidak mirip dengan salah satu citra pada *database* maka *output* menjadi *Buzzer ON*, LCD tampil “tidak mirip”, dan SMS akan dikirim sebagai indikasi adanya orang tidak dikenal telah terdeteksi oleh sistem.

Pada Gambar 5 dapat dilihat bahwa program dimulai dengan menginisialisasi pin yang digunakan pada Arduino dan variabel yang digunakan pada program tersebut. Kemudian setelah proses inisialisasi selanjutnya program akan membaca nilai dari *sensor* cahaya dan jarak. Berikutnya *output* atau hasil keluaran dari program ini adalah nilai *sensor* cahaya dalam *lux* dan nilai *sensor* jarak dalam cm.

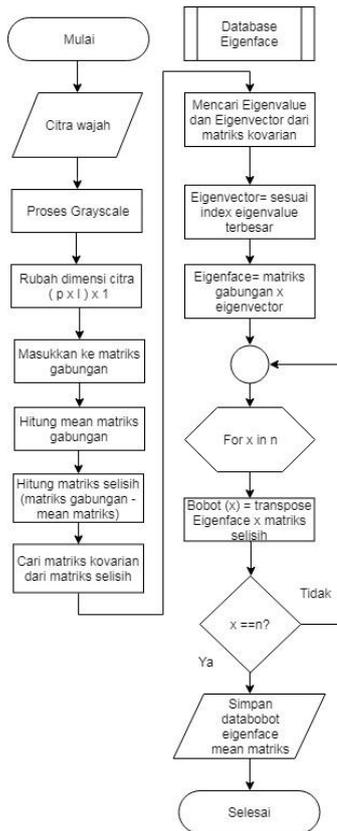
Pada Gambar 6 dapat dilihat perancangan deteksi wajah bertujuan untuk mendeteksi citra wajah dan menyimpan hasil deteksi citra wajah tersebut dengan dimensi yang sama. Ukuran citra wajah yang digunakan pada penelitian ini yaitu berukuran 92x112 piksel. Pada perancangan deteksi wajah di sistem ini menggunakan bantuan *library* OpenCV yaitu *library haarcascade frontalface.xml*. Deteksi wajah dimulai dengan inisialisasi variabel kemudian membaca *input* nilai dari *sensor* jarak atau ultrasonik dan cahaya yang berasal dari Arduino. Apabila nilai dari *sensor* jarak sama dengan 40 cm dan nilai dari *sensor* cahaya lebih besar dari sama dengan 5 *lux* maka *webcam* akan mengambil gambar, namun bila kedua kondisi diatas tidak terpenuhi maka program akan membaca nilai dari 2 *sensor* tersebut sampai nilai dari kedua *sensor* itu terpenuhi sesuai kondisi diatas. Jika gambar telah diambil oleh *webcam* selanjutnya gambar tersebut akan diproses untuk mencari objek wajah dengan



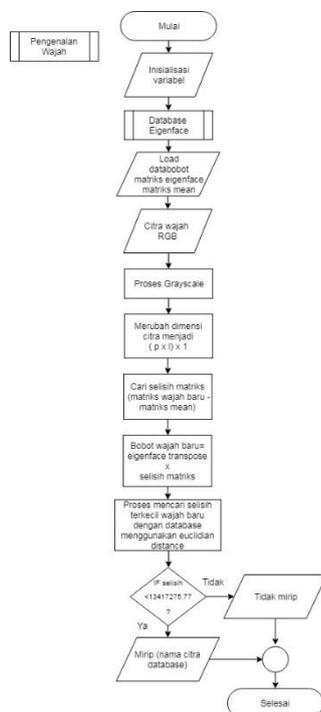
Gambar 6. Diagram Alir Deteksi Wajah

menggunakan *library Haarcascade*. Bila objek wajah tidak terdeteksi pada gambar, maka program akan mengulang untuk mengambil gambar sampai didapatkan objek wajah pada gambar tersebut. Setelah objek wajah terdeteksi pada gambar, selanjutnya dilakukan proses *cropping* yang dimana tujuan dari proses ini adalah untuk mendapatkan gambar hanya pada area objek wajah yang terdeteksi saja. Selanjutnya setelah proses *cropping*, citra wajah disimpan dengan ukuran 92 x 112 piksel dengan *format .pgm*.

Pada Gambar 7 merupakan diagram alir *Database Eigenface*. Dapat dilihat bahwa fungsi dari *Database Eigenface* dimulai dengan *input* citra wajah yang berasal dari *database* kemudian citra wajah tersebut diubah dari RGB ke *Grayscale*. Selanjutnya mengubah dimensi dari setiap citra tersebut menjadi satu dimensi. Gabungkan setiap citra berdimensi satu tersebut menjadi sebuah matriks gabungan. Hitung *mean* matriks, kemudian hitung matriks selisih. Cari matriks kovarian dari matriks selisih. Selanjutnya cari *Eigenvalue* dan *Eigenvector* dari matriks kovarian. Pilih nilai *Eigenvector* berdasarkan indeks nilai *Eigenvalue* terbesar. Cari *Eigenface* dengan cara mengalikan matriks gabungan dengan *Eigenvector*. Selanjutnya untuk mencari bobot masing-masing citra pada



Gambar 7. Diagram Alir Database Eigenface



Gambar 8. Diagram Alir Pengenalan Wajah

database yaitu dengan cara mengalikan matriks *transpose Eigenface* dengan matriks selisih. Langkah terakhir adalah menyimpan nilai bobot tersebut ke *database*.

Kemudian Pada Gambar 8 merupakan diagram alir pengenalan wajah. Pertama



Gambar 9. Implementasi *Prototype* Pengenalan Wajah

program dimulai dengan inisialisasi variabel yang digunakan. Kemudian *load* databobot, matriks *Eigenface*, dan matriks *mean* untuk digunakan pada perhitungan yang dihasilkan dari subprogram *Database Eigenface*. Selanjutnya citra wajah yang sudah diproses sebelumnya dimasukkan dan dikonversi dari RGB ke *grayscale*. Kemudian citra *grayscale* berupa matriks tersebut dirubah dimensinya menjadi 1x (panjang x lebar) citra. Cari selisih matriks yaitu (matriks wajah baru - matriks *mean*). Selanjutnya untuk mencari bobot wajah baru yaitu dengan cara mengalikan matriks *transpose Eigenface* dari *database* dengan matriks selisih yang sudah dihitung sebelumnya. Kemudian setelah hasil bobot wajah baru muncul. Selanjutnya mencari selisih terkecil bobot citra wajah baru dengan bobot citra wajah pada *database* dengan menggunakan metode *Euclidian Distance*. Jika bobot selisih terkecil < 13417275.77 maka program akan menghasilkan keluaran mirip (nama citra). Namun apabila bobot selisih terkecil diatas ambang batas tersebut maka program akan menghasilkan keluaran tidak mirip.

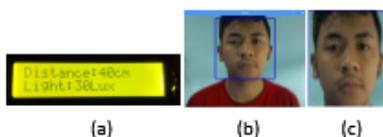
2.3 Implementasi Sistem

Dari perancangan yang telah dibuat sebelumnya, dapat dilihat pada Gambar 9. yang merupakan hasil implementasi *prototype* dari sistem pengenalan wajah yang berupa rangkaian sistem yang telah dirancang sebelumnya. Bahan yang digunakan sebagai wadah atau tempat adalah akrilik dengan ketebalan sekitar 2mm.

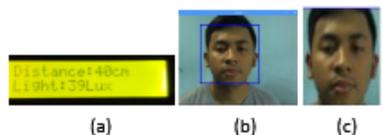
3. PENGUJIAN DAN ANALISIS

3.1 Pengujian Deteksi Wajah

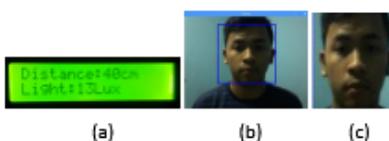
Pengujian akurasi dalam mendeteksi wajah sangat penting karena proses ini akan sangat berpengaruh terhadap akurasi pengenalan wajah nantinya. Pada pengujian ini dilakukan proses



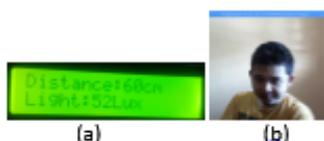
Gambar 10. Contoh Hasil Pengujian Deteksi Wajah Pagi Hari



Gambar 11. Contoh Hasil Pengujian Deteksi Wajah Siang Hari



Gambar 12. Contoh Hasil Pengujian Deteksi Wajah Malam Hari



Gambar 13. Contoh Hasil Pengujian Deteksi Wajah yang Tidak Terdeteksi

deteksi citra yang ditangkap. Proses deteksi dilakukan dengan menggunakan *haar-classifier* untuk mendapatkan objek wajah pada citra. Pada pengujian ini menggunakan 8 wajah orang yang berbeda-beda dan dilakukan pada waktu pagi, siang, malam dengan intensitas cahaya (pagi = 25-37 lux, siang = 40-55 lux, dan malam = 10-21 lux) dan juga dengan jarak 40 cm, 60 cm, 80 cm, dan 100 cm.

Gambar 10-12 adalah citra gambar yang dapat dideteksi objek wajah sedangkan Gambar 13 adalah citra gambar yang tidak terdeteksi. Hal ini dikarenakan cahaya yang tidak merata pada permukaan wajah.

Dari Tabel 2 dapat dilihat rata-rata total hasil akurasi dari pengujian deteksi wajah dengan jarak berbeda-beda. Pada jarak 40 cm total akurasinya adalah 100%, pada jarak 60 cm sebesar 91,67%, pada jarak 80 cm sebesar 95,83%, dan pada jarak 100 cm sebesar 87,5%. Sehingga didapatkan jarak akurasi deteksi wajah terbaik yaitu pada jarak 40 cm. Jarak tersebut digunakan pada pengujian selanjutnya yaitu pengenalan wajah.

Tabel 2. Rata-Rata Total Hasil Akurasi Pengujian Deteksi Wajah Dengan Jarak Berbeda-Beda

Jarak (cm)	Jumlah Benar		Jumlah Salah		Total rata-rata akurasi (%)
	Database	Luar database	Database	Luar database	
40 cm	12	12	0	0	100%
60 cm	11	11	1	1	91,67%
80 cm	11	12	1	0	95,83%
100 cm	11	10	1	2	87,5%

Tabel 3. Hasil Pengujian Pengenalan Wajah Pada Jarak 40 cm (objek dalam *database*)

No	Citra Uji	Nilai Jarak (cm)	Nilai Cahaya (lux)	Mirip	Sistem mengenali Dengan Benar
1	nanda_pagi.pgm	40 cm	30 lux	nanda	ya
2	isak_pagi.pgm	40 cm	28 lux	isak	ya
3	yosal_pagi.pgm	40 cm	35 lux	yosal	ya
4	herin_pagi.pgm	40 cm	34 lux	herin	ya
5	nanda_siang.pgm	40 cm	39 lux	nanda	ya
6	isak_siang.pgm	40 cm	41 lux	isak	ya
7	yosal_siang.pgm	40 cm	46 lux	yosal	ya
8	herin_siang.pgm	40 cm	49 lux	herin	ya
9	nanda_malam.pgm	40 cm	13 lux	tidak mirip	tidak
10	isak_malam.pgm	40 cm	14 lux	tidak mirip	tidak
11	yosal_malam.pgm	40 cm	17 lux	tidak mirip	tidak
12	herin_malam.pgm	40 cm	20 lux	tidak mirip	tidak

Tabel 4. Hasil Pengujian Pengenalan Wajah Pada Jarak 40 cm (objek diluar *database*)

No	Citra Uji	Nilai Jarak (cm)	Nilai Cahaya (lux)	Mirip	Sistem mengenali Dengan Benar
1	billy_pagi.pgm	40 cm	36 lux	tidak mirip	ya
2	edo_pagi.pgm	40 cm	33 lux	tidak mirip	ya
3	madya_pagi.pgm	40 cm	35 lux	isak.pgm	tidak
4	prasetyo_pagi.pgm	40 cm	33 lux	tidak mirip	ya
5	billy_siang.pgm	40 cm	49 lux	tidak mirip	ya
6	edo_siang.pgm	40 cm	49 lux	herin.pgm	tidak
7	madya_siang.pgm	40 cm	51 lux	tidak mirip	ya
8	prasetyo_siang.pgm	40 cm	21 lux	tidak mirip	ya
9	billy_malam.pgm	40 cm	21 lux	tidak mirip	ya
10	edo_malam.pgm	40 cm	18 lux	tidak mirip	ya
11	madya_malam.pgm	40 cm	19 lux	tidak mirip	ya
12	prasetyo_malam.pgm	40 cm	19 lux	tidak mirip	ya

3.2 Pengujian Pengenalan Wajah

Pengujian kedua yaitu pengujian pengenalan wajah berdasarkan intensitas cahaya (pagi = 25-37 lux, siang = 40-55 lux, dan malam = 10-21 lux). Pengujian ini dilakukan apabila nilai *Eigenface* dan bobot dari citra wajah yang digunakan pada *database* sudah dihasilkan. Pengujian dilakukan dengan menggunakan citra wajah orang yang sama pada *database* dan orang

```

pi@raspberrypi: ~/Desktop/tugasakhir
File Edit Tabs Help
v..service files
sistem on.py:64: RuntimeWarning: This channel is already i
/v.. Use GPIO.setwarnings(False) to disable warnings.
(GPIO.setup(11,GPIO.OUT)
nanda_pagi.pgm
##
bobot citra baru -16697693.8682
#####
bobot yosall.pgm 7534835.22501
selisih citra baru dengan yosall.pgm = 24232529.8932
#####
bobot nandal.pgm -30114969.6346
selisih citra baru dengan nandal.pgm = 13417275.7665
#####
bobot isak1.pgm -1754745.59002
selisih citra baru dengan isak1.pgm = 14942948.2782
#####
bobot herin1.pgm 24334879.9997
selisih citra baru dengan herin1.pgm = 41032573.8078
#####
selisih paling minimal 13417275.7665
mirip nandal.pgm
waktu Pengenalan Wajah 0.078831911087
    
```

Gambar 14. Contoh Hasil Pengujian Pengenalan Wajah yang dikenali

```

ido_pagi.pgm
##
bobot citra baru -53561300.0605
#####
bobot yosall.pgm 7534835.22501
selisih citra baru dengan yosall.pgm = 61096135.2855
#####
bobot nandal.pgm -30114969.6346
selisih citra baru dengan nandal.pgm = 23446330.4259
#####
bobot isak1.pgm -1754745.59002
selisih citra baru dengan isak1.pgm = 51806554.4705
#####
bobot herin1.pgm 24334879.9997
selisih citra baru dengan herin1.pgm = 77896180.8602
#####
selisih paling minimal 23446330.4259
Tidak mirip dan jarak paling mendekati adalah nandal.pgm
waktu Pengenalan Wajah 0.135311126709
ada orang masuk pada jam:15:06:22 Tue,11 Sep 2018
kirim sms berhasil
    
```

Gambar 15. Contoh Hasil Pengujian Pengenalan Wajah yang tidak dikenali

diluar *database* dengan jarak 40 cm dikarenakan jarak 40 cm menghasilkan hasil akurasi deteksi wajah terbaik.

Hasil pengujian pengenalan wajah (objek dalam *database*) dapat dilihat pada Tabel 3. Akurasi pengenalan wajah sudah baik pada waktu pagi dan siang hari. Akan tetapi pada malam hari sistem tidak dapat mengenali sama sekali karena faktor cahaya yang sangat berpengaruh pada akurasi pengenalan wajah. Didapatkan hasil akurasinya sebesar 66,67%. Sedangkan pada Tabel 4. merupakan hasil pengujian pengenalan wajah (objek diluar *database*). Akurasi pengenalan wajah sudah baik. Terdapat 2 citra orang diluar *database* yang dikenali oleh sistem. Hal tersebut terjadi karena faktor warna kulit dari objek yang bersangkutan dan intensitas cahaya saat pengambilan gambar. Didapatkan hasil akurasinya sebesar 83,33%. Total akurasi pengenalan wajah seluruhnya adalah sebesar 75%.

3.3 Pengujian Akurasi Integrasi Software Dengan Hardware

Pengujian ini merupakan pengujian dari *output* sistem apakah *output* sesuai dengan *input* yang diberikan atau tidak berdasarkan dari hasil

perhitungan selisih bobot citra uji dengan citra *database*. Pengujian ini dilakukan dengan menggunakan 24 citra yang dimana 12 citra orang dalam *database* dan 12 citra orang diluar *database* pada jarak 40 cm dan pada waktu pagi, siang, dan malam hari.

Akurasi integrasi *software* dengan *hardware* sudah baik. *Buzzer* aktif dan SMS terkirim apabila citra yang masuk sebagai *input* tidak dikenali oleh sistem atau melebihi ambang batas yang sudah ditentukan. Secara keseluruhan tingkat presentasi *error* dari integrasi *software* dengan *hardware* yaitu sebesar 0%..

3.4 Pengujian Waktu Rata-Rata Pengenalan Wajah

Pengujian terakhir adalah pengujian waktu rata-rata pengenalan wajah. Pengujian ini dilakukan untuk mengetahui waktu rata-rata komputasi pengenalan wajah. Pengujian ini dilakukan dengan menggunakan 24 citra yang dimana 12 citra orang dalam *database* dan 12 citra orang diluar *database* pada jarak 40 cm dan pada waktu pagi, siang, dan malam hari.

Hasil rata-rata waktu komputasi pengenalan wajah adalah sebesar 0.11536 detik.

4. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan , terdapat beberapa kesimpulan yaitu:

1. Sistem dapat mendeteksi dan mengenali wajah seseorang dalam *database* maupun diluar *database* dengan baik.
2. Akurasi deteksi wajah berdasarkan intensitas cahaya (pagi = 25-37 lux, siang = 40-55 lux, dan malam = 10-21 lux) dengan jarak berbeda beda yaitu pada jarak 40cm sebesar 100%, pada jarak 60cm sebesar 91,67%, pada jarak 80cm sebesar 95,83%, dan pada jarak 100cm sebesar 87,5%. Sehingga didapatkan akurasi deteksi wajah terbaik yaitu pada jarak 40 cm. Beberapa citra yang tidak dapat dideteksi dikarenakan faktor cahaya yang tidak merata pada permukaan wajah dan posisi wajah yang tidak tegak menghadap ke kamera.
3. Hasil akurasi pengujian pengenalan wajah berdasarkan intensitas cahaya (pagi = 25-37 lux, siang = 40-55 lux, dan malam = 10-21 lux) dengan jarak 40cm yaitu pada waktu pagi hari dengan objek dalam *database* sebesar 100% dan 75% untuk objek diluar *database*. Pada waktu siang hari dengan

objek dalam *database* sebesar 100% dan 75% untuk objek diluar *database*. Sedangkan pada waktu malam hari dengan objek dalam *database* sebesar 0% dan 100% untuk objek diluar *database*. Sehingga didapatkan total akurasi pengenalan wajah seluruhnya yaitu sebesar 75%. Beberapa citra yang salah saat dikenali oleh sistem dikarenakan faktor intensitas cahaya yang sangat berpengaruh terhadap akurasi pengenalan wajah.

4. Integrasi sistem *software* dengan *hardware* sudah baik. Presentasi *error* dari integrasi sistem adalah 0%. Sistem dapat mengeluarkan *output* sesuai yang diharapkan. Jika ada orang tidak dikenal terdeteksi oleh sistem maka sistem akan mengirimkan sms serta *buzzer* akan berbunyi, namun apabila sistem mendeteksi dan mengenali citra wajah tersebut berada di *database* maka *buzzer* tidak akan aktif dan sms tidak akan terkirim.
5. Waktu rata-rata komputasi yang dibutuhkan dalam mengenali wajah adalah sebesar 0.11536 detik.

5. DAFTAR PUSTAKA

- Bruce, V., & Young, A. (2018). *Wiley Online Library*. Retrieved 06 26, 2018, from <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2044-8295.1986.tb02199.x>
- Budiawan, I., & Andriana. (2014). Pengujian Pengenalan Wajah Menggunakan Raspberry Pi. *J.Oto.Ktrl.Inst (J.Auto.Ctrl.Inst)*, 6 (2), 10. Retrieved 06 13, 2018, from journals.itb.ac.id/index.php/joki/article/viewFile/3899/1926
- Nicco, & Fahruzy, I. (2014). Rancang Bangun Sistem Biometrik Pengenalan Wajah Menggunakan Principal Component Analysis. 6, no. 1, 2014, 64-71, 8.
- Susanto, B. M., Purnomo, F. E., & Fahmi, M. F. (2017). Sistem Keamanan Pintu Berbasis Pengenalan Wajah Menggunakan Metode Fisherface. *Jurnal Ilmiah INOVASI*, 17 No.1, 10. Retrieved 06 13, 2018, from <https://publikasi.polije.ac.id/index.php/jii/article/view/464/401>
- Turk, M. A., & Pentland, A. P. (1991). Face Recognition Using Eigenfaces. 6. Retrieved 06 16, 2018, from <https://www.cs.ucsb.edu/~mturk/Papers>

/mturk-CVPR91.pdf

- Wasista, S., D, B. S., & Putra, S. A. (2011). Sistem Pengenalan Wajah Pada Mesin Absensi Mahasiswa. *Sistem Pengenalan Wajah Pada Mesin Absensi Mahasiswa*, 5. Retrieved 06 16, 2018, from <http://repo.pens.ac.id/1439/>
- Yanto, A. N., Hurriyatul Fitriyah, S. M., & Mochammad Hannats Hanafi I., S. (2016). Implementasi Algoritma Pengenalan Wajah Eigenface Pada Intel Galileo Untuk Sistem Keamanan Ruang. 10.