

# Real-Time Face Recognition For Smart Home Application

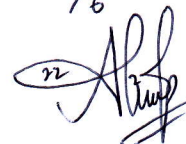
Usep Syaikal Arifin<sup>1</sup>, Almira Budiyan<sup>2</sup>

Jurusan Teknik Elektro, Universitas Islam Indonesia  
Jl Kaliurang KM 14.5 Yogyakarta, Indonesia

<sup>1</sup>14524063@students.uui.ac.id

<sup>2</sup>almira.budiyan@uui.ac.id

20/6/19



(Almira - B)

**Abstrak**—Wajah merupakan ciri fisiologis yang paling mudah digunakan untuk membedakan identitas setiap individu. Wajah merupakan salah satu aplikasi teknologi biometrika yang menggunakan ciri-ciri fisik yang khas dari tubuh seseorang sebagai pembeda antara satu orang dengan orang lainnya dan memiliki karakteristik diantaranya tidak dapat hilang, sulit dipalsukan dan tidak mungkin lupa karena keberadaannya melekat pada manusia itu sendiri. Sehingga dilakukan perancangan sistem pengenalan wajah sebagai sistem kendali buka tutup pintu menggunakan metode *eigenface* dan *euclidean distance*. Metode *eigenface* berfungsi untuk menghitung *eigenvalue* dan *eigenvector* yang akan digunakan sebagai fitur dalam melakukan pengenalan. Metode *eigenface* mempunyai komputasi yang sederhana dan cepat. Metode jarak *euclidean* digunakan untuk mencari jarak dengan data fitur yang telah didapat dan jarak terkecil adalah hasilnya. Hasil pengujian *face detection system* menunjukkan bahwa objek dapat deteksi dengan batasan jarak objek dengan kamera adalah 40 cm sampai 400 cm dan objek tidak memakai baju warna putih dan warna kulit. Hasil *face recognition system* menggunakan *euclidean distance* didapatkan akurasi 80% dan 90%.

**Kata kunci**—*face detection system*, *face recognition system*, *euclidean distance*, metode *eigenface*.

## I. PENDAHULUAN

Keamanan adalah hal penting dalam kehidupan sehari-hari. Keamanan dapat dimulai dari hal yang paling utama yaitu rumah. Sistem keamanan rumah yang kurang baik mengakibatkan rumah menjadi sasaran pencurian atau tindak kejahatan lainnya[1]. Pintu menjadi hal yang paling disoroti dalam ruang lingkup sistem keamanan ruangan, hal tersebut karena fungsi pintu sebagai akses utama untuk masuk ruangan[2]. Sistem keamanan pada kunci rumah saat ini kebanyakan merupakan sistem keamanan manual berupa gembok atau kunci konvensional[1]. Penggunaan kunci konvensional kurang praktis karena pemilik rumah harus membawa kunci tersebut jika berpergian dan sering kali pemilik rumah lupa bahkan kehilangan kunci[3]. Sehingga dengan berkembangnya teknologi, sistem keamanan dapat dilakukan dengan menggunakan alat elektronik sebagai pengganti sistem keamanan kunci konvensional. Teknologi biometrika merupakan teknologi yang dalam kerjanya menggunakan ciri-ciri fisik yang khas dari tubuh seseorang sebagai pembeda antara satu orang dengan orang lainnya dan memiliki karakteristik diantaranya tidak dapat hilang, sulit dipalsukan dan tidak mungkin lupa karena keberadaannya

melekat pada manusia itu sendiri[4]. Teknologi biometrika mempunyai banyak macam diantaranya sidik jari, retina, rambut, suara dan wajah[5]. Wajah merupakan ciri fisiologis yang paling mudah digunakan untuk membedakan identitas setiap individu[6]. Teknologi sebuah sistem pengamanan untuk masuk ke sebuah rumah telah banyak dikembangkan seperti menggunakan kartu, *password*. Metode tersebut masih memiliki kekurangan seperti keterbatasan manusia dalam mengingat benda dan mengingat kombinasi angka yang menyebabkan tidak dapat diaksesnya pintu tersebut[7]. Sistem keamanan rumah konvensional dinilai kurang efektif dan efisien. Hal tersebut disebabkan karena pengguna diharuskan memasukan kunci terlebih dahulu kemudian memutar kunci kearah tertentu sehingga pintu dapat dibuka. Oleh karena itu alangkah baiknya jika sistem pengenalan wajah dapat digunakan untuk sistem keamanan rumah. Sistem keamanan rumah menggunakan wajah memiliki kelebihan seperti wajah tidak dapat di duplikasi, wajah manusia berbeda-beda dan lebih efisien karena pemilik rumah hanya perlu mengarahkan wajah ke kamera untuk membuka pintu. Oleh karena itu penulis membuat sistem pengenalan wajah sebagai sistem kendali buka tutup pintu ruangan. Penggunaan wajah sebagai kunci untuk masuk ke dalam rumah diharapkan dapat meningkatkan tingkat keamanan menjadi lebih tinggi dikarenakan wajah tidak dapat diduplikasi dan berpindah tangan.

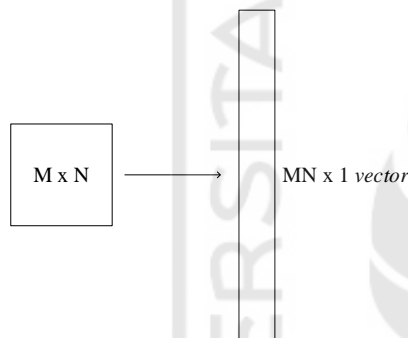
## II. TINJAUAN PUSTAKA

Penelitian mengenai *face recognition* menggunakan metode *eigenface* pernah dilakukan oleh Liton Chandra Paul. Kesimpulan jurnal tesis Liton adalah sistem berhasil mengenali wajah manusia dengan berbagai orientasi. Namun dalam penelitian tersebut tidak dijelaskan bagaimana sistem mengambil wajah dari camera yang digunakan sehingga hanya bagian wajah saja yang terdeteksi[8]. Sedangkan dalam penelitian Mariappan dkk algoritma dalam proses *face detection* adalah *image threshold*, *filtering*, *edge detection*, *morphological* dan *circle detection* (Mariappan, 2014). Hasil dari penelitian tersebut sistem bekerja dengan baik untuk latar belakang gambar yang berbeda-beda. Wajah dapat dideteksi bahkan jika orang tersebut berada pada jarak yang berbeda-beda dari kamera[9]. Kemudian dalam penelitian Figen disimpulkan bahwa untuk meningkatkan tingkat keberhasilan pengenalan wajah menggunakan metode *eigenface*, dapat diperkaya dengan informasi segitiga wajah dan memperbesar *database*[10]. Sedangkan Malkauthekar mengatakan dalam

hasil penelitiannya bahwa metode *manhattan distance* lebih baik daripada *euclidean distance*[11]. Peneliti Pooja mengatakan *mahalanobis distance* mempunyai kinerja lebih baik dari pada *euclidean distance* dalam hal *face recognition*[12]. Kemudian Fan dkk dalam penelitiannya mengatakan bahwa fitur yang sangat populer dalam *machine learning* dan *patern recognition* adalah *Principal Component Analysis (PCA)*[13]. Sedangkan sistem pengenalan wajah atau *face recognition* terdiri dari 2 bagian. Pertama adalah bagian *hardware* sedangkan yang ke dua adalah *software*. *Hardware* atau perangkat keras yang digunakan adalah kamera, laptop dan mikrokontroler. *Software* atau perangkat lunak yang digunakan adalah *LabVIEW (Laboratory Virtual Instrument Engineering Workbench)*.

#### A. Algoritma Eigenface

Asumsikan bahwa setiap citra  $\Gamma(x, y)$  memiliki resolusi  $M \times N$  *pixel*. Kemudian ubah setiap citra menjadi citra abu-abu (*grayscale*), setelah itu mengubah citra ke dalam bentuk matrik. Sehingga secara umum algoritma *eigenface* adalah mengubah matrik citra abu-abu berukuran  $M \times N$  menjadi citra *vector* berukuran  $MN \times 1$  seperti Gambar 1 berikut:



Gambar 1. Mengubah citra resolusi  $M \times N$  *pixel* menjadi  $MN \times 1$  *vector*

Setiap citra dinyatakan dengan notasi  $\Gamma_i$ .  $\Gamma_i$  adalah *vector* yang merepresentasikan citra wajah.  $\Gamma_i$  adalah *vector* berukuran  $MN \times 1$  yang berkorespondensi dengan citra wajah  $M \times N$  *pixel*. Kemudian membentuk *training set* citra wajah ( $\Gamma_1 \Gamma_2 \Gamma_3 \Gamma_4 \Gamma_5, \dots \dots \dots \Gamma_m$ ) untuk pelatihan. Citra wajah harus mempunyai resolusi serta *align* yang sama. Menghitung *average face vector* ( $\Psi$ ) dengan persamaan berikut:

$$\Psi = \frac{1}{n} \sum_{i=1}^n \Gamma_n \quad (1)$$

dengan nilai:

- $\Psi$  = *average face vector* / rata-rata *vector* citra wajah
- $\Gamma_n$  = *vector* citra wajah ke -  $n$  ukuran  $MN \times 1$
- $n$  = jumlah citra wajah

Menghitung selisih antara *vector* wajah  $\Gamma_i$  dengan *average face vector*  $\Psi$ . Selisih tersebut disimbolkan  $\Phi$ .

$$\Phi_i = \Gamma_i - \Psi \quad (2)$$

dengan nilai:

- $\Phi_i$  = nilai ke- $i$  selisih antara *vector* wajah ( $\Gamma_i$ ) dengan *average face vector* ( $\Psi$ )

Menghitung matrik kovarian  $C$  sebagai berikut:

$$C = \frac{1}{n} \sum_{i=1}^n \Phi_i \Phi_i^T = AA^T \quad (MN \times MN) \quad (3)$$

dengan nilai:

- $C$  = matrik kovarian
- $A$  =  $[\Phi_1 \Phi_2 \Phi_3 \Phi_4 \Phi_5, \dots \dots \Phi_n]$  matrik  $MN \times M$
- $A^T$  = matrik *transpose* dari matrik  $A$

Menghitung *eigenvector* ( $u_i$ ) dari matrik kovarian  $C$ . Berdasarkan Persamaan (3) matrik kovarian  $C$  memiliki dimensi  $MN \times MN$ . Ukuran tersebut sangat besar sehingga membuat keadaan menjadi tidak praktis (B.Utomo, 2011)[14]. Misal citra dengan resolusi  $256 \times 256$  *pixel* maka proses komputasi menghitung matrik  $65.356 \times 65.356$  dan  $65.356$  *eigenvalue*. Nilai tersebut merupakan jumlah data yang sangat besar dan tidak efisien untuk proses komputasi. Sehingga diperlukan beberapa trik untuk menghitung *eigenvalue*nya (B.Utomo, 2011)[14]. Solusi untuk menghitung *eigenvector* ( $u_i$ ) dari matrik  $AA^T$  yang memiliki ukuran yang sangat besar adalah sebagai berikut:

Menghitung matrik  $L$  menggunakan Persamaan (4) berikut:

$$L = A^T A \quad (4)$$

Menghitung *eigenvector* ( $v_i$ ) dari matrik  $L$ . Matrik  $L$  memiliki resolusi  $M \times M$ , kemudian menghitung *eigenvector* ( $v_i$ ) dari  $A^T A$  dengan menyelesaikan persamaan berikut:

$$A^T A v_i = \mu_i v_i \quad (5)$$

dengan nilai:

- $v_i$  = *eigenvector* dari matrik  $L$
- $\mu_i$  = *eigenvector* dari matrik kovarian  $C$

Kesimpulannya untuk menemukan *eigenvector* dan *eigenvalue* dari matrik kovarian  $C$  cukup dengan mencari *eigenvector* dan *eigenvalue* dari matrik  $L$ .

Mencari *eigenvector*  $v_i$  dari matrik  $L$  dengan *eigenvalue*  $\neq 0$ .

Mencari  $M$  *eigenface* dari matrik  $L$  dengan persamaan berikut:

$$u_i = \sum_{i=1}^n v_i \Phi_i \quad (6)$$

dengan nilai:

- $u_i$  = nilai *eigenface* ke -  $i$

Memproyeksikan citra *dataset* pada dasarnya adalah mencari bobot atau nilai kombinasi linier terbaik untuk merepresentasikan wajah (B.Utomo, 2011)[14]. Kombinasi linier dapat dihitung dengan mengalikan matrik selisih antar citra ( $\Phi$ ) dengan nilai *eigenface* ( $u_i$ ). nilai bobot ( $\omega$ ) dapat dinyatakan sebagai persamaan berikut:

$$\omega_i = u_i^T \Phi_i \quad (7)$$

$$\omega_i = u_i^T (\Gamma_i - \Psi) \quad (8)$$

dengan nilai:



$\omega_i$  = nilai bobot ke - i

Sehingga matrik bobot ( $\Omega$ ) dapat dirumuskan sebagai berikut:

$$\Omega_i^T = [\omega_1, \omega_2, \omega_3 \dots \dots \omega_K] \quad (9)$$

### B. Proyeksi Citra Datatest

Citra *datatest* adalah citra keluaran dari *face detection*. Memproyeksikan citra *datatest* adalah mengubah citra *datatest*  $M \times N$  menjadi citra *datatest*  $MN \times 1$ . *Vector* citra *datatest* tersebut disimbolkan  $\Gamma_{test}$ . Sehingga persamaan yang digunakan untuk memproyeksikan citra *datatest* adalah sebagai berikut:

$$\omega_{test} = u_i^T (\Gamma_{test} - \Psi) \quad (10)$$

$$\Omega_{test}^T = [\omega_1, \omega_2, \omega_3 \dots \dots \omega_K] \quad (11)$$

dengan nilai:

$\omega_{test}$  = nilai bobot citra *datatest*

$\Omega_{test}^T$  = transpose matrik bobot citra *datatest*

$\Gamma_{test}$  = vector citra *datatest* berukuran  $MN \times 1$

### C. Aplikasi Citra Menggunakan Euclidean Distance

*Euclidean Distance* dihitung dengan persamaan berikut:

$$d(\Omega_{test}, \Omega_i) = \sqrt{\sum_{i=1}^n (\Omega_{test} - \Omega_i)^2} \quad (12)$$

dengan nilai:

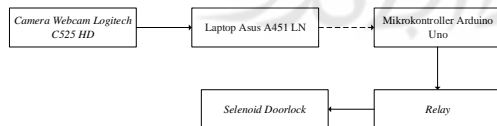
$d(\Omega_{test}, \Omega_i)$  = nilai jarak *euclidean*

Hasil dari penelitian ini akan diukur dengan menghitung nilai *error* dari hasil klasifikasi yang benar dengan semua data uji [15]. Tingkat akurasi ini dihitung dengan menggunakan rumus :

$$Error = \frac{\text{jumlah klasifikasi yang benar}}{\text{jumlah semua data uji}} \times 100\% \quad (13)$$

## III. METODE PENELITIAN

Perancangan sistem ini bertujuan menggunakan pengenalan wajah untuk keamanan rumah. Pintu rumah dapat terbuka jika wajah terdeteksi sebagai anggota penghuni rumah. Sistem keamanan rumah akan disimulasikan menggunakan *LabVIEW* 2017 dan mikrokontroler *Arduino Uno*. Aktuator pembuka kunci rumah akan disimulasikan menggunakan *Solenoid Doorlock 12 Volt DC*. Diagram blok dari perancangan perangkat keras seperti pada Gambar 2 berikut:

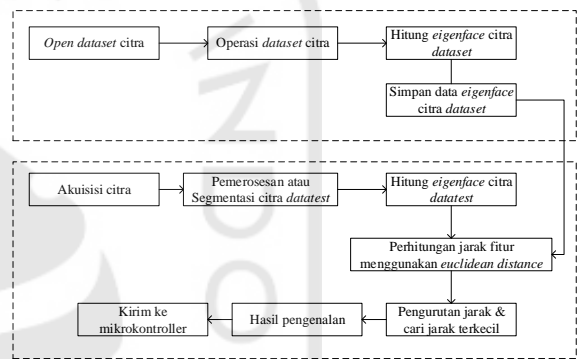


Gambar 2. Blok diagram perangkat keras

Kamera yang digunakan adalah *Logitech C525 portable HD Webcam 720p 30fps 8MP (Mega Pixel)*. Kamera tersebut terhubung dengan laptop *Asus A451 LN* menggunakan

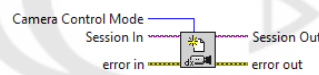
perpanjangan kabel USB 1,5 meter. Pada laptop citra di proses menggunakan *LabVIEW software*. Selain itu laptop berfungsi untuk memonitor proses dan mengubah parameter yang dibutuhkan. Mikrokontroler *Arduino Uno* menerima data dalam bentuk data serial atau *string* dari *LabVIEW software*. Data dikirim menggunakan *Bluetooth* bawaan laptop yaitu *Qualcomm Atheros AR3012 Communications*. Sedangkan pada mikrokontroler *Arduino Uno* data diterima oleh *Bluetooth HC 05*. Data serial yang dikirim dari *LabVIEW* digunakan untuk menentukan kondisi *output port* digital *Arduino Uno* yang terhubung ke *relay*. Kondisi *port* digital *Arduino Uno* tersebut bisa *High* atau *Low*. *Relay* berfungsi sebagai saklar untuk *Solenoid Doorlock*. Kondisi *relay ON* menandakan bahwa *supply* tegangan 12 VDC untuk *Solenoid Doorlock* terhubung. Sedangkan kondisi *relay OFF* menandakan *supply* tegangan 12 VDC untuk *Solenoid Doorlock* tidak terhubung.

Secara keseluruhan proses pengenalan wajah menggunakan metode *eigenface* pada *LabVIEW* digambarkan sebagai berikut:

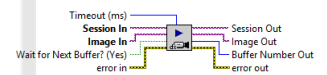


Gambar 3. Blok diagram perancangan perangkat lunak

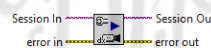
Pada *LabVIEW* proses akuisisi citra menggunakan blok diagram *IMAQdx Open Camera.vi*. Blok *IMAQdx Open Camera.vi* ditunjukkan pada Gambar 4. Blok tersebut berfungsi untuk mengkasas kamera yang terhubung ke laptop. Kamera yang diakses bisa kamera internal laptop atau eksternal laptop. Blok diagram *IMAQdx Open Camera* kemudian dihubungkan dengan blok diagram *IMAQdx Configure Grab*. Blok *IMAQdx Configure Grab* ditunjukkan pada Gambar 6. Fungsi dari blok tersebut untuk mengatur resolusi kamera.



Gambar 4. *IMAQdx Open Camera.vi*



Gambar 5. *IMAQdx Grab.vi*



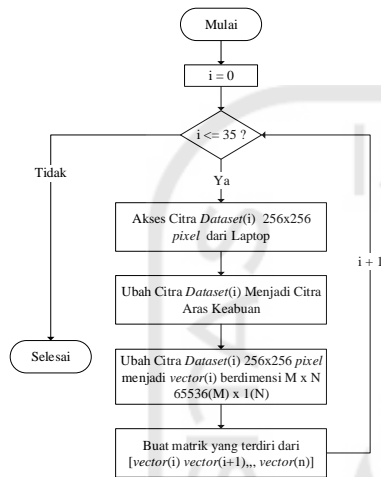
Gambar 6. *IMAQdx Configure Grab.vi*



Gambar 7. *IMAQdx Close Camera.vi*

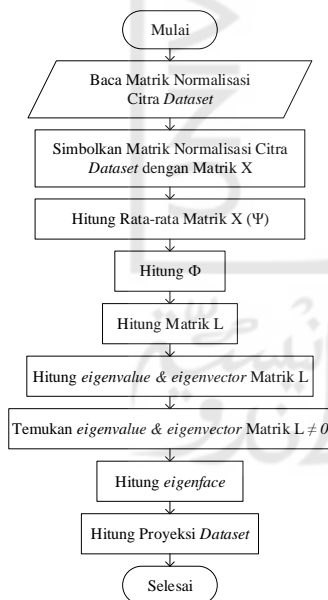
Berdasarkan Gambar 8 jumlah *dataset* yang digunakan berjumlah 36 buah citra. Masing-masing citra *dataset* memiliki resolusi 256x256 *pixel* dengan format *.jpg*. Citra *dataset* tersebut berisi citra dari orang-orang yang di ijinakan untuk masuk ke kamar. Proses pengambilan citra *dataset* menggunakan kamera *smartphone*. Kemudian resolusinya diubah dari *default* bawaan *smartphone* menjadi resolusi

256x256 pixel menggunakan software photoshop. Setelah di edit menggunakan software photoshop citra dataset disimpan dalam format jpg dan RGB. Pada proses ini dilakukan perubahan citra dari citra RGB menjadi citra aras keabuan (grayscale). Sehingga nilai dari masing-masing titik pada tiap pixel nya memiliki nilai 0 sampai 255 saja. Kemudian citra dataset tersebut diubah menjadi vector yang memiliki dimensi  $M = 65536$  dan  $N = 1$ . Kemudian masing-masing vector tersebut dihimpun menjadi sebuah matrik. Matrik tersebut dalam penelitian ini diberi nama matrik Normalisasi Citra Dataset.



Gambar 8. Flowchart proses open dataset citra

Berdasarkan Gambar 9 proses yang terjadi pada kondisi operasi dataset citra adalah proses pembelajaran citra dataset. Pelajari citra dataset adalah proses mengubah 36 citra dataset ke ruang dataset menggunakan algoritma eigenface. Flowchart dari proses tersebut terdapat pada Gambar 9 sebagai berikut:

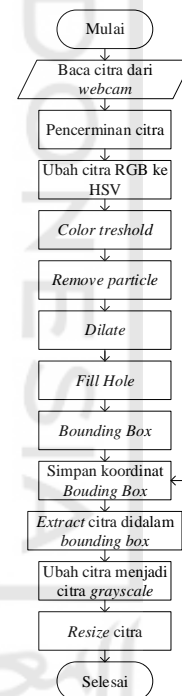


Gambar 9. Flowchart proses operasi citra dataset

Matrik X adalah matrik yang berasal dari himpunan training set citra wajah ( $\Gamma_1 \Gamma_2 \Gamma_3 \Gamma_4 \Gamma_5, \dots \dots \Gamma_m$ ) yang

memiliki dimensi  $M = 65536$  dan  $N = 1$ . Matrik X memiliki dimensi 65536 baris dan 36 kolom. Kemudian average face vector ( $\Psi$ ) memiliki 65536 elemen. Persamaan yang digunakan untuk mendapatkan average face vector adalah Persamaan (1). Nilai selisih antara matrik X dan average face vector ( $\Psi$ ) atau ( $\Phi$ ) dinyatakan dalam sebuah matrik A yang memiliki dimensi 65536 baris dan 36 kolom. Persamaan yang digunakan untuk mendapatkan matrik A adalah Persamaan (2). Matrik L diperoleh dengan mengalikan matrik A transpose dengan matrik A. matrik A memiliki dimensi 65536 baris dan 36 kolom. Sedangkan matrik A transpose memiliki dimensi 36 baris dan 65536 kolom. Maka dimensi untuk matrik L adalah 36 baris dan 36 kolom sesuai Persamaan (4). Kemudian untuk menemukan eigenvalue dan eigenvector dari matrik L menggunakan code mathscript eig(L) pada LabVIEW 2017. Sintak eig(L) memiliki arti mencari nilai eigenvalue dan eigenvector dari matrik L. Kemudian eigenvector yang digunakan adalah eigenvector yang memiliki eigenvalue tidak sama dengan nol.

Face detection adalah proses mengambil bagian wajah sampai leher saja dari keseluruhan frame citra. Proses face detection harus dilakukan sebelum proses membandingkan citra dataset dengan citra dataset. Flowchart proses face detection ditunjukkan Gambar 10 berikut.

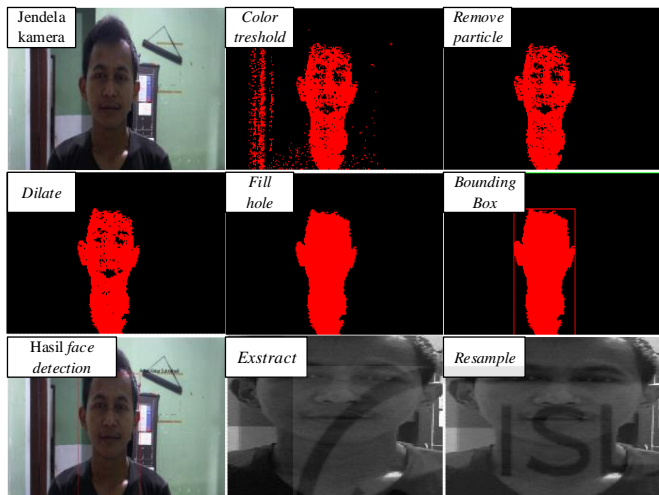


Gambar 10. Flowchart face detection

Proses color threshold bertujuan untuk mendapatkan bagian wajah sampai leher objek. Color threshold dilakukan dengan cara mengatur parameter hue minimum, hue maksimum, saturation minimum, saturation maksimum, value minimum dan value maksimum dari blok IMAQ ColorThreshold VI pada LabVIEW.

Nilai parameter terbaik untuk masing-masing parameter didapatkan dari blok IMAQ ColorHistogram VI pada LabVIEW. Hasil dari color threshold ditunjukkan Gambar 11

yang mana Gambar 11 menunjukkan bahwa proses *color threshold* meloloskan citra berwarna kulit dengan nilai biner 1.

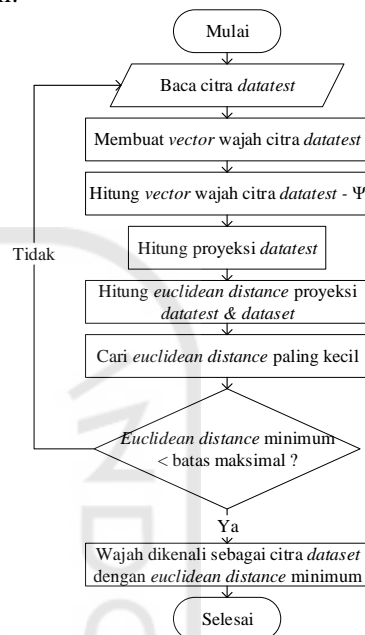


Gambar 11. Proses *face detection*

Proses *remove particle* menggunakan blok *IMAQ RemoveParticle VI* pada *LabVIEW*. Proses *remove particle* bertujuan untuk memperbaiki kualitas citra sehingga didapatkan citra yang hanya terdiri dari bagian wajah sampai leher objek. Proses *dilate* menggunakan blok *IMAQ Morphology VI* pada *LabVIEW*. Proses *dilate* bertujuan untuk mendapatkan efek pelebaran terhadap *pixel* bernilai 1. Proses *fill hole* menggunakan blok *IMAQ FillHole VI* pada *LabVIEW*. Proses *fill hole* bertujuan menambah kualitas citra dengan cara mengisi partikel berlubang dengan nilai 1. Proses *bounding box* bertujuan untuk mendeteksi objek dan mengetahui koordinat objek. Proses *bounding box* menggunakan blok *IMAQ CountObjek 2 VI* pada *LabVIEW*. Koordinat objek yang didapatkan digambarkan dengan *bounding box* berwarna merah disekitar objek seperti ditunjukkan pada Gambar 11. *Bounding box* tersebut kemudian diletakan pada citra hasil pembacaan *webcam* seperti yang terdapat pada Gambar 11 bagian hasil *face detection*. Proses *extract* bertujuan untuk mengambil bagian objek yang terdapat pada *bounding box*. Proses *extract* menggunakan blok *IMAQ Extract VI*. Kemudian proses yang terakhir adalah proses *resize* citra menggunakan blok *IMAQ Resample VI*. Proses *resize* bertujuan untuk mengubah resolusi citra menjadi resolusi citra yang bersesuaian dengan citra *dataset* yaitu  $256 \times 256$ .

*Face recognition* adalah proses mengidentifikasi wajah sehingga dapat ditentukan apakah wajah dikenali atau tidak. Prinsip dasar dari *face recognition* adalah menghitung jarak *euclidean* antara proyeksi citra *datatest* dengan proyeksi citra *dataset*. *Flowchart* proses *face recognition* ditunjukkan Gambar 12. Baca citra *datatest* adalah proses mengambil data keluaran dari blok *IMAQ Resample VI* seperti pada Gambar 11. Citra yang ditunjukkan oleh Gambar 11 bagian *resample* adalah contoh citra yang dijadikan citra *datatest*. Proses membuat *vector* wajah adalah proses mengubah citra seperti pada Gambar 11 bagian *resample* menjadi *vector* berukuran  $65536 \times 1$  jika resolusi citra *datatest* adalah  $256 \times 256$  *pixel*. Proses selanjutnya adalah mencari selisih antara *vector* citra *datatest* dengan *average face vector* ( $\Psi$ ) yang diperoleh sesuai

Persamaan (1). Proses menghitung proyeksi *dataset* menggunakan Persamaan (7). Kemudian proses yang terakhir adalah menghitung *euclidean distance* antara nilai proyeksi *datatest* dengan proyeksi *dataset* menggunakan Persamaan (12). Hasil akhir dari perhitungan tersebut adalah 36 nilai *euclidean distance*. Kemudian dari 36 nilai tersebut dicari nilai yang paling kecil.

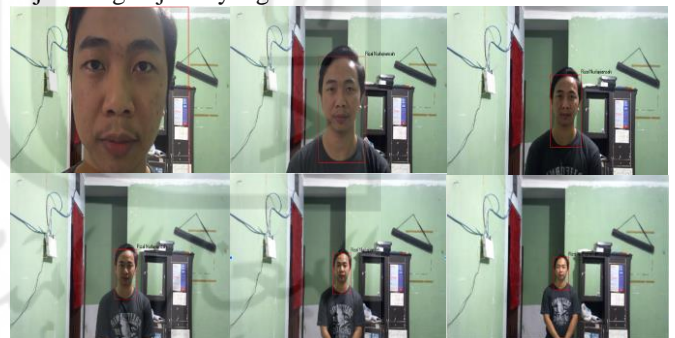


Gambar 12. *Flowchart face recognition*

#### IV. HASIL DAN ANALISIS

##### A. Pengujian Face Detection System dengan Variasi Jarak Objek dengan Kamera

Pengujian ini bertujuan untuk mengetahui kemampuan *face detection system* mendeteksi bagian wajah sampai leher objek dengan jarak yang bervariasi.



Gambar 13. Pengujian *face detection system* dengan variasi jarak

Pengujian ini dilakukan dengan kondisi pencahayaan tetap. Berdasarkan Gambar 13 *face detection system* mampu mendeteksi objek dengan variasi jarak dari 40 sampai 400 cm. Pada awalnya tujuan pengujian ini untuk mengetahui jarak terdekat dan terjauh dari *face detection system*. Karena keterbatasan tempat tidak diperoleh batas jarak maksimum antara jarak objek dengan kamera. Jarak terdekat objek dengan

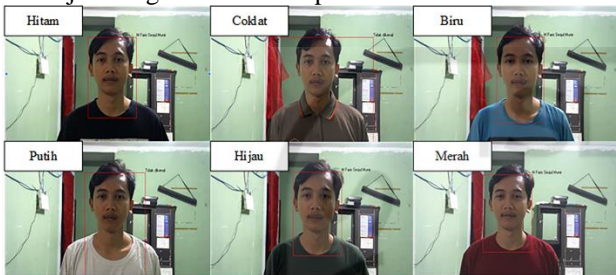


kamera adalah 40 cm. jika jarak kurang dari 40 cm maka seluruh layar tertutupi oleh bagian muka objek.

### B. Pengujian Face Detection System dengan Variasi Warna Pakaian Objek

Pengujian ini bertujuan untuk mengetahui kemampuan *face detection system* mendeteksi bagian wajah sampai leher objek dengan variasi warna pakaian objek.

Pengujian ini bertujuan untuk mengetahui apakah warna pakaian yang dikenakan oleh objek dapat mempengaruhi kemampuan *face detection system* dalam mendeteksi objek. Pengujian ini dilakukan dengan kondisi pencahayaan tetap dan jarak objek dengan kamera tetap.



Gambar 14. Pengujian *face detection system* dengan variasi warna baju objek

Berdasarkan Gambar 14 warna yang identik dengan kulit yaitu coklat membuat *face detection system* tidak bekerja dengan baik. Selain warna coklat warna putih juga membuat *face detection system* tidak bekerja dengan baik sehingga citra *datatest* tidak dikenali. Hal ini menunjukkan warna yang sangat kontras dengan kulit membuat *face detection system* tidak bekerja dengan baik. Tingkat keberhasilan percobaan *face detection system* menggunakan variasi warna pakaian adalah 60%.

### C. Pengujian Face Detection System dengan Variasi Aksesoris

Pengujian ini bertujuan untuk mengetahui kemampuan *face detection system* mendeteksi bagian wajah sampai leher objek dengan variasi aksesoris yang dikenakan oleh objek. Pengujian ini bertujuan untuk mengetahui apakah aksesoris yang dikenakan oleh objek dapat mempengaruhi kemampuan *face detection system* dalam mendeteksi objek. Pengujian ini dilakukan dengan kondisi pencahayaan tetap.



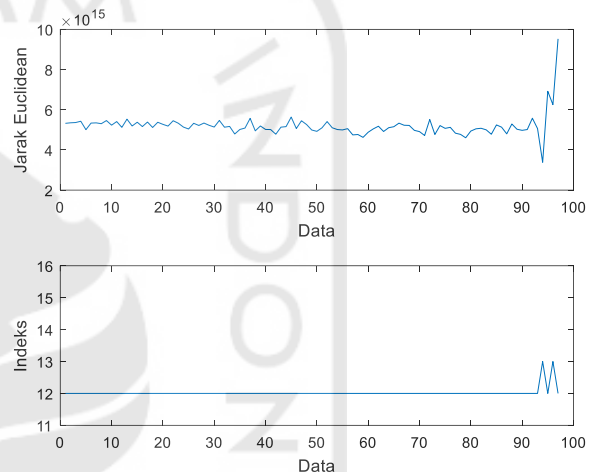
Gambar 15. Pengujian *face detection system* dengan variasi aksesoris

Berdasarkan Gambar 15 *face detection system* mampu mendeteksi objek dengan baik. Aksesoris yang digunakan adalah aksesoris yang lazim dipakai. Tingkat keberhasilan dari percobaan *face detection system* menggunakan variasi

aksesoris adalah 100%. Persamaan yang digunakan dalam menentukan tingkat keberhasilan pengujian sistem menggunakan Persamaan (13).

### D. Pengujian Ketahanan Face Recognition System

Pertama pengujian ini bertujuan untuk mengetahui *face recognition system* mempertahankan nilai jarak *euclidean*. Kedua pengujian ini bertujuan untuk mengetahui fluktuatif nilai jarak *euclidean*. Ketiga pada penelitian ini kamera yang digunakan memiliki fitur *autofocus* sehingga ada kemungkinan nilai jarak *euclidean* dan indeksnya bisa berubah ketika objek bergerak. Percobaan D menggunakan objek 3. Pada percobaan D objek 3 dalam kondisi tetap. Kondisi tetap adalah objek 3 tidak mengalami perubahan posisi apapun dan tidak melakukan pergerakan sedikitpun. Kondisi tetap dipertahankan selama 1 menit. Data yang diperoleh adalah 97 data seperti gambar berikut:

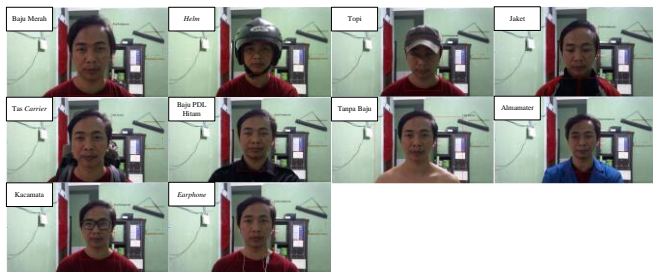


Gambar 16. Grafik nilai jarak *euclidean* dan indeks objek 3

Berdasarkan Gambar 16 nilai jarak *euclidean* objek 3 memiliki fluktuatif yang sangat rendah. Rata-rata nilai jarak *euclidean* adalah  $5,19333E+15$ . Batas maksimal nilai jarak *euclidean* objek 3 adalah  $1,31076E+16$ . Gambar 16 menunjukkan hasil pembacaan indeks pada percobaan objek 3. Indeks yang terbaca adalah 12 dan 13. Indeks yang menunjukkan bahwa objek adalah objek 3 memiliki nilai 12 sampai 17. Berdasarkan Gambar 16 pada percobaan D objek 3 dikenali sebagai objek 3. Sehingga kesimpulannya adalah percobaan D memiliki tingkat keberhasilan 100%.

### E. Pengujian dengan Objek dikenali Sebagai Dataset

Pengujian dilakukan dengan beberapa kondisi variabel tetap. Variabel tersebut adalah kondisi cahaya dan jarak objek kamera yang dianggap tetap. Percobaan ini dilakukan dengan menggunakan 6 objek yang terdaftar sebagai *dataset*. Percobaan ini dilakukan sebanyak 10 kali percobaan pada masing-masing objek yang terdaftar dengan variasi aksesoris yang digunakan objek.



Gambar 17. Hasil Pengujian dengan objek 2 dikenali sebagai *dataset*

Objek 2 dikenali jika hasil jarak *euclidean* dibawah nilai batas maksimalnya. Batas maksimal jarak *euclidean* adalah  $1,06112E+16$ . Objek 2 dikenali sebagai objek 2 jika indeks jarak *euclidean* bernilai 7 sampai 11. Berdasarkan Gambar 17 percobaan dengan objek 2 memiliki tingkat keberhasilan 80%. Jarak *euclidean* pada percobaan objek 2 nilainya  $2,94E+15$  sampai dengan  $7,14E+15$ . Kesalahan mengenali objek ketika objek 2 menggunakan tas *Carrier* dan tidak memakai baju. Pada percobaan objek 2 menggunakan Tas *Carrier* objek 2 tidak terdeteksi pada bagian wajah saja tapi mengalami pelebaran. Sehingga nilai jarak *euclidean* yang terbaca adalah  $7,79E+16$  dengan indeks jarak *euclidean* bernilai 14. Indeks 14 menunjukkan objek dikenali sebagai objek 3. Sedangkan objek 3 nilai batas maksimalnya adalah  $1,31076E+16$ . Kemudian pada percobaan objek 2 tidak memakai baju karena sistem mendeteksi bukan hanya bagian wajah saja namun sampai ke bagian badan objek. Sehingga nilai jarak *euclidean* yang terbaca adalah  $2,45E+16$  dengan indeks jarak *euclidean* bernilai 32. Indeks 32 menunjukkan objek dikenali sebagai objek 6. Sedangkan objek 6 nilai batas maksimalnya adalah  $1,14447E+16$ .

Pengujian selanjutnya dilakukan dengan beberapa kondisi variabel tetap. Variabel tersebut adalah kondisi cahaya dan jarak objek kamera yang dianggap tetap. Percobaan ini dilakukan dengan menggunakan 2 objek yang tidak terdaftar sebagai *dataset*. Percobaan ini dilakukan sebanyak 9 kali percobaan pada masing-masing objek yang tidak terdaftar dengan variasi aksesoris yang digunakan objek.



Gambar 18. Hasil pengujian dengan objek 7 tidak dikenali sebagai *dataset*

Berdasarkan Gambar 18 akurasi percobaan dengan objek 7 tidak dikenali sebagai *dataset* adalah 100%. Jarak *euclidean* pada percobaan objek 7 nilainya  $1,40E+16$  sampai dengan  $2,57E+16$ . Objek 7 dikenali sebagai objek 4 (objek yang terdaftar pada *dataset*) dengan indeks jarak *euclidean* bernilai 19 dan 20. Sedangkan nilai batas maksimal jarak *euclidean* untuk objek 4 adalah  $1,2452E+16$ .

## V. KESIMPULAN

Berdasarkan percobaan A sampai E kesimpulan dari penelitian *face detection system* dan *face recognition system* adalah sebagai berikut:

1. *Face detection system* dapat bekerja dengan baik dengan beberapa batasan yaitu interval jarak objek dengan kamera 40 cm sampai 400 cm, tidak memakai baju yang berwarna putih dan warna yang identik dengan kulit objek dan tidak menggunakan aksesoris yang menutupi bagian wajah dan leher.
2. *Face recognition system* bekerja dengan baik dengan batasan yaitu objek tidak melakukan banyak pergerakan ketika wajah dihadapkan pada kamera.
3. *Face recognition system* mampu membedakan objek yang dikenali dan yang tidak dikenali dengan akurasi 80% dan 90%.

## DAFTAR PUSTAKA

- [1] A. R. Gifari Alim Prakasa, "Prototype Sistem Kunci Pintu Berbasis," 2017.
- [2] V. Raja, "Pengenalan Wajah Sebagai Sistem Kendali Buka Tutup Pintu Ruang Dengan Metode Support Vector Machine," 2018.
- [3] E. H. H. guntoro Yoyo Somantri, "Rancang Bangun Magnetic Door Lock Menggunakan Keypad Dan Solenoid Berbasis Mikrokontroler Arduino Uno," *Electrans*, vol. 12, no. 1, pp. 39–48, 2013.
- [4] M. Miftah, J. T. Informatika, U. Dian, N. Semarang, P. Wajah, and T. Face, "Pengamanan Laptop Menggunakan Pengenalan Wajah Berbasis Triangle Face," *J. Appl. Intell. Syst. UDINUS*, vol. 1, no. 1, pp. 22–35, 2016.
- [5] S. Dhian, "Analisa Algoritma Sistem Keamanan Poin Minutiae Pada Hp Compact 2210B Notebook Pc," *J. Univ. Gunadarma*, vol. 6, no. 01, pp. 18–21, 2012.
- [6] B. Septian, A. Wijayanto, F. Utamingrum, and I. Arwani, "Face Recognition Untuk Sistem Pengaman Rumah Menggunakan Metode HOG dan KNN Berbasis Embedded," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 3, pp. 2774–2781, 2019.
- [7] F. Muhammad, "Sistem Keamanan Akses Pintu Masuk Menggunakan Face Recognition Berbasis Raspberry Pi 3," 2018.
- [8] Liton Chandra Paul and Abdulla Al Sumam, "Face Recognition Using Principal Component Analysis Method," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 1, no. 9, pp. 2278–1323, 2012.
- [9] M. Mariappan, "A LabVIEW Design for Frontal and Non- Frontal Human Face Detection System in Complex Background," *Appl. Mech. Mater.*, vol. 490–491, no. 1, pp. 1259–1266, 2014.

- [10] M. Çar and F. Özen, "A Face Recognition System Based on Eigenfaces Method," *Procedia Technol*, vol. 1, no. 1, pp. 118–123, 2012.
- [11] M. D. Malkauthekar, "Analysis of euclidean distance and manhattan distance measure in face recognition," *Third Int. Conf. Comput. Intell. Inf. Technol*, vol. 1, no. 4, pp. 3–7, 2013.
- [12] A. Dagar, M. T. E. C. E. Hod, C. Engineering, and F. Nagar, "Face Recognition Using Mahalanobis Distance In MATLAB," *Int. J. Technol. Res. Eng.*, vol. 5, no. 8, pp. 3499–3503, 2018.
- [13] Z. Fan, M. Ni, M. Sheng, Z. Wu, and B. Xu, "Principal component analysis integrating mahalanobis distance for face recognition," *Proc. - 2013 2nd Int. Conf. Robot. Vis. Signal Process. RVSP 2013*, pp. 89–92, 2013.
- [14] B. Utomo, "Matematika Eigenface Menggunakan Metrik Euclidean," in *Seminar Nasional Matematika dan Pendidikan Matematika2*, 2011, vol. 6, no. 3, pp. 44–52.
- [15] A. H. Mulyadi, L. Talangko, and A. Lawi, "Analisis Komponen Utama Dan Jarak Euclidean," *Progr. Stud. Stat. FMIPA Univ. Hasanuddin*, vol. 9, no. 6, pp. 1–14, 2014.

