JITE (Journal of Informatics and Telecommunication Engineering)

Available online <http://ojs.uma.ac.id/index.php/jite> DOI : 10.31289/jite.v3i2.3132

Face Recognition based Feature Extraction using Principal Component Analysis (PCA)

Muhammad Zulfahmi Nasution^{1)*}

Prodi Sistem Komputer, Fakultas Sains dan Teknologi, Universitas Pembangunan Pancabudi
Medan, Indonesia

*Corresponding Email: fhm.nasution@dosen.pancabudi.ac.id

Abstrak

Wajah manusia merupakan entitas yang memiliki fitur semantik. Pendeteksian wajah merupakan tahap awal sebelum dilakukan proses pengenalan wajah. Teknik pengenalan wajah merupakan proses identifikasi berdasarkan gambaran raut wajah. Salah satu pendekatan ekstraksi fitur untuk teknik pengenalan wajah adalah metode *Principal Component Analysis* (PCA). Metode PCA digunakan untuk menyederhanakan fitur dan karakteristik wajah sehingga diperoleh proporsi yang mampu mewakili karakteristik wajah asli. Tujuan penelitian ini adalah melakukan konstruksi pola wajah yang tersimpan pada database citra digital. Proses konstruksi pola dan pengenalan wajah dimulai dari objek berupa citra wajah, deteksi sisi, konstruksi pola hingga dapat menentukan kesamaan pola wajah untuk dilanjutkan sebagai pengenalan wajah. Dalam penelitian ini telah dirancang sebuah program untuk menguji beberapa sampel data wajah yang tersimpan pada database citra digital hingga dapat memberikan suatu kesamaan pola wajah yang sedang diamati dan pengenalannya menggunakan PCA.

Kata Kunci: Ekstraksi Fitur, Reduksi Fitur, PCA, Pola Wajah, Pengenalan Wajah.

Abstract

The human face is an entity that has semantic features. Face detection is the first step before face recognition. Face recognition technique is an identification process based on facial features. One feature extraction approach for facial recognition techniques is the Principal Component Analysis (PCA) method. The PCA method is used to simplify facial features and characteristics in order to obtain proportions that are able to represent the characteristics of the original face. The purpose of this research is to construct facial patterns stored in a digital image database. The process of pattern construction and face recognition starts from objects in the form of face images, side detection, pattern construction until it can determine the similarity of face patterns to proceed as face recognition. In this research, a program has been designed to test some samples of face data stored in a digital image database so that it can provide a similarity in the face patterns being observed and its introduction using PCA

Keywords: Feature Extraction, Feature Reduction, PCA, Face Pattern, Face Recognition.

How to Cite: Nasution, M.Z. (2020). Face Recognition based Feature Extraction using Principal Component Analysis. *JITE (Journal of Informatics and Telecommunication Engineering)*. 3 (2): 182-191

I. PENDAHULUAN

Wajah manusia merupakan sebuah entitas yang memiliki fitur semantik. Pendeteksian wajah merupakan tahap awal sebelum dilakukan proses pengenalan wajah. Teknik pengenalan wajah merupakan proses identifikasi berdasarkan gambaran raut wajah. Algoritma yang digunakan dalam pengenalan wajah memang cukup bervariasi, Namun ada tiga tahap dasar yang sama, yaitu tahap deteksi wajah (*face detection*), Tahapan ekstraksi komponen-komponen wajah (*facial features extraction*), serta tahap pengenalan wajah (*face-recognition*). Pengenalan wajah manusia dapat diimplementasikan kedalam banyak aplikasi salah satunya adalah pencarian database individu pada database citra digital. Citra wajah dapat digunakan untuk menganalisa identitas seseorang.

Pendeteksian wajah (*face detection*) merupakan tahapan awal sebelum dilakukannya proses pengenalan wajah (Kresimir D & Mislav G, 2007). Salah satu pendekatan yang digunakan untuk pengenalan wajah adalah menggunakan pendekatan *Principal Component Analysis*. Pendekatan berbasis PCA biasanya mencakup dua fase: pelatihan dan klasifikasi. Dalam fase pelatihan, *eigenface* dibentuk berasal data sampel pelatihan

menggunakan PCA dan gambar wajah pelatihan dipetakan ke *eigenface* untuk diklasifikasikan. Dalam fase klasifikasi, wajah input diproyeksikan ke dalam *eigenface* yang sama dan diklasifikasikan oleh kategori yang sesuai (Eleyan, A. & Demirel, H. 2005).

PCA memberikan hasil yang baik ketika diterapkan pada data saling berkorelasi. Dalam penelitian ini, PCA digunakan untuk mengidentifikasi pola dalam kumpulan data, menemukan persamaan dan perbedaannya diantara masing-masing atribut (Nasution, MZ. 2018).

Karena PCA berfungsi sebagai model yang kuat untuk menganalisis data. Data asli dan rata-rata data asli dipilih. Matriks kovarians dihitung dimana hasilnya digunakan dalam menghitung vektor eigen dengan nilai eigen tertinggi dipilih sebagai komponen utama dari data (Jolliffe, I.T. 2002).

Metode PCA digunakan untuk mengkonstruksi pola input wajah digital menggunakan teknik propagasi dalam pengenalan wajah. Dalam proses konstruksi pola dan pengenalan wajah mulai dari objek berupa citra wajah, deteksi sisi, konstruksi pola hingga dapat menentukan suatu karakteristik baru untuk dilanjutkan sebagai pengenalan wajah.

Tujuan dari penelitian ini adalah melakukan konstruksi pola wajah yang tersimpan pada database citra digital. Proses konstruksi pola dan pengenalan wajah dimulai dari objek berupa citra wajah, deteksi sisi, konstruksi pola hingga dapat menentukan kesamaan pola wajah untuk dilanjutkan sebagai pengenalan wajah. Dalam penelitian ini telah dirancang sebuah program untuk menguji beberapa sampel data wajah yang tersimpan pada database citra digital sehingga dapat memberikan sebuah kesamaan pola wajah yang sedang diamati dan pengenalannya menggunakan PCA. Sehingga membantu mempercepat proses pengenalan wajah.

II. STUDI PUSTAKA

A. Pengolahan Citra

Pengolahan citra digital (*Digital Image Processing*) adalah penggunaan algoritma komputer untuk melakukan pengolahan citra pada citra digital. Citra yang dimaksud adalah gambar diam maupun gambar bergerak. Sedangkan pengertian *digital* berarti pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan fungsi *continue* dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka citra harus

dipresentasikan secara numerik dengan nilai-nilai diskrit. Representasi dari fungsi *continue* menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat direpresentasikan oleh matriks 2 dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana diagonal antara kolom dan baris disebut *pixel* atau elemen terkecil dari sebuah citra.

B. Principal Component Analysis

PCA merupakan kombinasi linear dari variabel awal yang secara geometris kombinasi linear ini merupakan sistem koordinat baru yang diperoleh dari rotasi system semula. Metode PCA sangat berguna digunakan jika data yang ada memiliki jumlah variabel yang besar dan memiliki korelasi antar variabelnya. Perhitungan dari *principal component analysis* didasarkan pada perhitungan nilai eigen dan vektor eigen yang menyatakan penyebaran data dari suatu dataset (Johnson, W.A. & Wichern, D.W. 2007).

Dengan menggunakan PCA, variabel yang tadinya sebanyak n variabel akan diseleksi menjadi k variabel baru yang disebut *principal component*, dengan jumlah k lebih sedikit dari n . Dengan hanya menggunakan k *principal component* akan menghasilkan nilai yang sama dengan menggunakan n variabel. Variabel hasil dari seleksi disebut *principal component* (Kotu, V. & Deshpande, B. 2015).

PCA digunakan untuk menjelaskan struktur matriks varians-kovarians dari suatu set variabel melalui kombinasi linier dari variabel-variabel tersebut. Secara umum *principal component* (PC) dapat berguna untuk seleksi fitur dan interpretasi variabel-variabel.

Skema konseptual yang diilustrasikan bagaimana PCA dapat membantu untuk menyederhanakan dimensi dari data melalui hipotesa dataset berjumlah m variabel dapat ditunjukkan pada gambar berikut :

Gambar 1. Model Konseptual PCA
(Sumber: Kotu & Deshpande, 2015)

Menurut (Jolliffe, 2002), Prosedur pengerjaan *Principal Component Analysis* bertujuan untuk menyederhanakan dan menghilangkan faktor atau indikator skrining yang kurang dominan dan kurang relevan tanpa mengurangi maksud dan tujuan dari data asli dari variabel acak x (matrik berukuran $n \times n$, dimana baris-baris yang berisi observasi sebanyak n dari variabel acak x) adalah sebagai berikut:

1. Menghitung matrik varians kovarian dari data observasi. Varians ($Var(x)$) dapat dihitung untuk menemukan penyebaran data dalam set data untuk menentukan penyimpangan data dalam set data sampel. Matrik Kovarian atau $Cov(x,y)$ merupakan matriks yang nilai-nilai kovariansi pada tiap *cell*-nya diperoleh dari sampel. Misalkan x dan y adalah variabel acak. (Jolliffe, 2002).

$$Var(x) = \sigma^2 = \frac{1}{n} \sum_{i=1}^n (Z_{ij} - \mu_j)^2 \quad (1)$$

$$Cov(x,y) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \mu_{xj})(y_{ij} - \mu_{yj}) \quad (2)$$

Dengan μ_x dan μ_y merupakan rata-rata (*mean*) sampel dari variabel x dan y , dimana variabel x_i dan y_i merupakan nilai observasi ke- i dari variabel x dan y . Dari data nilai yang ada, maka diperoleh matrik kovarian $n \times n$.

2. Mencari *eigenvalues* dan *eigenvector* dari matrik kovarian yang telah diperoleh yaitu: Nilai *eigen* dan vektor *eigen* untuk matriks kovarians dihitung. Nilai *eigen* yang dikomputasi kemudian ditransformasikan (*rotasi orthogonal varimax*) menggunakan persamaan berikut :

$$\text{Det}(A - \lambda I) = 0 \quad (3)$$

dimana:

A = matrik $n \times n$

λ = nilai *eigenvalue*

I = matriks identitas (matriks persegi

dengan elemen diagonal utama bernilai 1 sedangkan elemen lain bernilai 0)

- Menentukan nilai proporsi *Principal Component* (proporsi *Principal Component* (%)) dengan persamaan :

$$PC (\%) = \frac{NilaiEigen}{VarianceCovarian} \times 100\% \quad (4)$$

- Menghitung bobot factor (*factor loading*) berdasarkan *eigenvector* dengan persamaan (Jhonson & Wichern, 2007):

$$Ax = \lambda x \quad (5)$$

Sehingga diperoleh kombinasi linear yaitu:

- $\lambda_1, \lambda_2, \lambda_3 \dots \lambda_n$ adalah *eigenvalue* matrik A
- $x_1, x_2, x_3 \dots x_n$ adalah *eigenvector* sesuai *eigenvalue*-nya (λ_n)

Persamaan *eigenvalue* & *eigenvector* merupakan *Eigen Value Decomposition* (EVD), dengan persamaan sebagai berikut:

$$AX = XD \quad (6)$$

$$A = X D X^{-1}$$

dimana:

A = matrik $n \times n$ yang memiliki *n eigenvalue* (λ_n)

D = *eigenvalue* dari *eigenvector*-nya

X = *eigenvector* dari matrik A

X^{-1} = invers dari *eigenvector* X

III. METODE PENELITIAN

Penelitian ini bertujuan untuk membangun sebuah aplikasi pengenalan wajah dengan menerapkan metode PCA sebagai ekstraksi fitur pada masukan citra wajah digital sehingga dapat memberikan visualisasi hasil subcitra termasuk informasi yang terikat dalam citra tersimpan.

Aplikasi yang telah dirancang memungkinkan untuk mampu membantu mempercepat proses pengenalan wajah berdasarkan input citra terhadap sejumlah citra digital yang telah tersimpan didalam database citra sehingga informasi yang terikat citra tersebut dapat dimanfaatkan dengan baik.

Untuk lebih rinci dan jelas dalam mendeskripsikan proses dalam penelitian ini maka akan dijelaskan tahapan demi tahapan. Adapun secara garis besar dapat dilihat pada skema gambar berikut:

Gambar 2. Tahapan Umum Penelitian

Berdasarkan Gambar 2, Tahapan umum penelitian terlihat bahwa terdapat proses pengumpulan data karakteristik citra, dapat diuraikan sebagai berikut :

1. Mengumpulkan data citra dan menyeleksi sesuai dengan persyaratan dalam batasan berupa *resizing window* menjadi maksimum $200 \times 200 \text{ pixels}$.
2. Menyusun dan mengelompokkan formula untuk setiap *item* proses, serta membuat rancangan relasi antar proses hingga integrasi proses dalam sistem pengenalan wajah.
3. Merancang dan membangun input-process-output berbasis CLI.

Tahapan selanjutnya adalah ekstraksi fitur menggunakan *Principal Component Analysis* (PCA), Ekstraksi fitur merupakan proses yang bertujuan untuk menentukan ciri dari suatu citra ekspresi wajah. Adapun tahapan ekstraksi fitur menggunakan PCA sebagai berikut:

1. Hitung rata-rata seluruh sampel data diperoleh dengan menggunakan persamaan:

$$\bar{x}_j = \frac{\sum_{ij=1}^n x_{ij}}{n} \quad (7)$$

2. *Adjusted data* (data yang telah disesuaikan) adalah hasil pengurangan dari setiap data dengan rata-rata setiap

data yang diperoleh dengan persamaan berikut:

$$\text{Adjusted data} = x_{ij} - \bar{x}_j \quad (8)$$

$$X' = \text{Adjusted Data}$$

3. Hitung matrik kovarian (c) dihitung dengan menggunakan persamaan berikut:

$$c = \frac{1}{M} X' X'^T \quad (9)$$

4. Hitung nilai eigen dan vektor eigen dari matrik kovarian dihitung dengan menggunakan persamaan karakteristik berikut ini:

$$c - \lambda I = 0$$

$$(c - \lambda I)v = 0 \quad (10)$$

Dimana c adalah matrik kovarian, I adalah matrik Identitas, λ adalah nilai *eigen* dan v adalah vektor *eigen*.

5. Hitung nilai *eigen* yang terbesar yang berkorespondensi terhadap nilai vektor *eigen* yang terbesar dipilih menjadi *Principal Component*. Vektor *eigen* yang disusun dari yang terbesar ke yang terkecil dipilih menjadi vektor fitur.

$$v = (eig_1, eig_2, eig_3 \dots eig_n) \quad (11)$$

6. Untuk mencari *Principal Component* dengan ' sebagai rata-rata.

$$PC = X' \times v \quad (12)$$

7. Langkah berikutnya untuk melakukan transformasi data untuk menghasilkan data PCA.

$$PCA \text{ data} = PC^T \times X'^T \quad (13)$$

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengumpulan Data Citra

Proses awal dalam membangun aplikasi pengenalan citra wajah adalah mengumpulkan data - data citra yang akan digunakan dan disimpan sebagai database citra guna menganalisis karakteristik citra tersebut. Oleh karena itu, data citra yang telah dikumpulkan kemudian akan dibagi menjadi data latih citra dan data uji citra. Citra wajah yang diperoleh dari hasil capture wajah menggunakan camera device. Adapun data citra yang diujikan dapat dilihat pada tabel berikut:

Tabel 1. Data Citra Fahmi

Tabel 2. Data Citra Syahril

B. Hasil Transformasi Citra Warna ke Citra Grayscale

Berikut adalah hasil transformasi citra warna ke citra *grayscale* sebagai tahapan awal menggunakan dukungan bahasa pemrograman *Python* versi 3.7.3 berikut:

```
import os
import cv2
import numpy as np

def unFlatten(vector, rows, cols):
    img = []
    cutter = 0
    while(cutter+cols<=rows*cols):
        try:
            img.append(vector[cutter:cutter+cols])
        except:
            img = vector[cutter:cutter+cols]
        cutter+=cols
    img = np.array(img)
    return img

w=200
h=200
face=['fahmi','syahril']
```

Berikut adalah tampilan proses pembacaan file *database* citra digital:

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (tags/v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\PENELITIAN\RISET\4 # JITE\Internal\PCA\pca_training.py =====
Read from: fahmi Directory
WIN_20191111_12_17_27_Pro.jpg
WIN_20191111_12_17_30_Pro.jpg
WIN_20191111_12_17_33_Pro.jpg
WIN_20191111_12_17_36_Pro.jpg
WIN_20191111_12_17_56_Pro.jpg
WIN_20191111_12_18_13_Pro.jpg
WIN_20191111_12_18_16_Pro.jpg
WIN_20191111_12_18_18_Pro.jpg
Read from: syahril Directory
WIN_20191024_07_17_10_Pro.jpg
WIN_20191024_07_17_11_Pro.jpg
WIN_20191024_07_17_12_Pro.jpg
WIN_20191024_07_17_13_Pro (2).jpg
WIN_20191024_07_17_13_Pro.jpg
WIN_20191024_07_17_14_Pro (2).jpg
WIN_20191024_07_17_14_Pro.jpg
WIN_20191024_07_17_15_Pro (2).jpg
WIN_20191024_07_17_15_Pro.jpg
WIN_20191024_07_17_16_Pro.jpg
WIN_20191024_07_17_18_Pro (2).jpg
WIN_20191024_07_17_18_Pro.jpg
WIN_20191024_07_17_19_Pro.jpg
WIN_20191024_07_18_20_Pro.jpg
WIN_20191024_07_18_21_Pro (2).jpg
WIN_20191024_07_18_21_Pro.jpg
WIN_20191024_07_18_22_Pro (2).jpg
WIN_20191024_07_18_22_Pro.jpg
WIN_20191024_07_18_23_Pro.jpg
>>>
```

Gambar 3. Proses Pembacaan File *Database* Citra

Berikut adalah tampilan hasil transformasi citra warna ke citra *grayscale* yaitu:

Tabel 2. Hasil Transformasi Citra Warna ke Grayscale

C. Hasil Ekstraksi Fitur Principal Component Analysis (PCA)

Data latih yang diperoleh dari hasil transformasi citra warna ke citra hasil *grayscale* dilakukan ke tahap berikutnya yakni proses ekstraksi fitur menggunakan pendekatan *Principal Component Analysis* (PCA). Berikut adalah kode program ekstraksi fitur menggunakan pendekatan *Principal Component Analysis* (PCA) menggunakan dukungan bahasa pemrograman *Python* versi 3.7.3 sebagai berikut:

input citra

```
for v in face:
    in_matrix = None
    imgcnt=0
    print ('Read from: ' + v + ' Directory ')
    for f in os.listdir(os.path.join('training/',v)):
        imgcnt+=1
        print(f)
```

baca citra as a gray level image

```
img = cv2.imread(os.path.join('training/',v, f),
cv2.IMREAD_GRAYSCALE)
img_resized = cv2.resize(img,(w,h))
```

resize ke w * h

```
vec = img_resized.reshape(w * h)
```

stack to form the matrix

try:

```
in_matrix = np.vstack((in_matrix, vec))
```

except:

```
in_matrix = vec
```

Proses Ekstraksi PCA

if in_matrix is not None:

```
mean,eigenvalues=cv2.PCACompute(in_matrix,
np.mean(in_matrix, axis=0).reshape(1,-1))
```

```
img = unFlatten(mean.transpose(), w, h)
```

```
cv2.imwrite('trained/pca_face_'+v+'.png',img)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

D. Hasil Pencocokan Fitur Wajah

Proses selanjutnya adalah mencocokkan fitur hasil pelatihan citra atau hasil ekstraksi fitur menggunakan PCA yang telah disimpan pada database citra dengan fitur citra uji menggunakan metode *Euclidean Distance* agar dilakukan pencocokkan fitur-fitur citra hasil *grayscale* sebelumnya. Berikut adalah kode program untuk proses pencocokan fitur menggunakan pendekatan *Euclidean Distance* menggunakan dukungan bahasa pemrograman *Python* versi 3.7.3 sebagai berikut:

```
import os
```

```
import cv2
```

```
import numpy as np
```

```
w=200
```

```
h=200
```

```
#euclidean distance
```

```
def dist(x,y):
```

```
    return np.sqrt(np.sum((x-y)**2))
```



```

inputImageToPredict= 'testfahmi.jpg'
def main():
img=cv2.imread(inputImageToPredict,cv2.IMR
EAD_GRAYSCALE)
img_resized = cv2.resize(img,(w,h))
imgBlurred=cv2.GaussianBlur(img_resized,
(5,5),                                0)
imgThresh=cv2.adaptiveThreshold(imgBlurred,
255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.T
HRESH_BINARY,11,2)
cv2.imshow('Test Image',imgThresh)
npaFlattenedImage = imgThresh.reshape((1, w
* h))
    print (npaFlattenedImage.shape)
    mean_to= npaFlattenedImage.mean()
    m=[]
    vowels=['fahmi','syahril']
    cnt=0
    for v in vowels:
print ('Read ' + v + ' mean image from directory
!')
f= 'trained/pca_face_'+v+'.png'
print(f)
img2 = cv2.imread(f, cv2.IMREAD_GRAYSCALE)
img_resized2 = cv2.resize(img2,(w,h))
imgBlurred2 = cv2.GaussianBlur(img_resized2,
(5,5), 0)
imgThresh2
=
cv2.adaptiveThreshold(imgBlurred2,
255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.T
HRESH_BINARY,11,2)
npaFlattenedImage2= imgThresh2.reshape((1,
w * h))
distance      =      dist(npaFlattenedImage,
npaFlattenedImage2)
        m.append(distance);
        print ('Euclidean Distance Array: ')
        print (m)
        print ('Min Distance: ')
        print (min(m))

```

```

print ('Array Position: ')
pos=m.index(min(m))
print (pos)
print ("The Face Recognized Is : ")
print (vowels[pos])
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Berikut adalah output dari proses akhir pengenalan wajah *Fahmi*:

```

Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: G:\PENELITIAN\UNPAB\1 # RISET INTERNAL\3\Laporan\Internal\PCA\prediksi_face.py
(1, 40000)
Read ade mean image from directory !
trained/pca_face_ade.png
Read fahmi mean image from directory !
trained/pca_face_fahmi.png
Read firhan mean image from directory !
trained/pca_face_firhan.png
Read ivan mean image from directory !
trained/pca_face_ivan.png
Read nanda mean image from directory !
trained/pca_face_nanda.png
Euclidean Distance Array:
[[110.34491379306978, 68.75318174455637, 111.57060544785082, 118.61703081766969, 112.09817126073021]]
Min Distance:
68.75318174455637
Array Position:
1
The Face Recognized Is :
fahmi

```

Gambar 4. Hasil Pengenalan Wajah *Fahmi*
Berikut adalah pengenalan wajah *Syahril*:

```

Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\PENELITIAN\RISET\4 # JITE\Internal\PCA\prediksi_face.py =====
(1, 40000)
Read ade mean image from directory !
trained/pca_face_ade.png
Read fahmi mean image from directory !
trained/pca_face_fahmi.png
Read firhan mean image from directory !
trained/pca_face_firhan.png
Read ivan mean image from directory !
trained/pca_face_ivan.png
Read nanda mean image from directory !
trained/pca_face_nanda.png
Read syahril mean image from directory !
trained/pca_face_syahril.png
Euclidean Distance Array:
[[121.69812596141005, 117.95338062132852, 119.33985084622823, 123.80630032433729, 119.41524190822544, 107.20541031123382]]
Min Distance:
107.20541031123382
Array Position:
5
The Face Recognized Is :
syahril

```

Gambar 5. Hasil Pengenalan Wajah *Syahril*

V. SIMPULAN

Metode PCA digunakan untuk mengkonstruksi pola input wajah digital menggunakan teknik propagasi dalam pengenalan wajah. Dalam proses konstruksi pola dan pengenalan wajah mulai dari objek berupa citra wajah, deteksi sisi, konstruksi pola hingga dapat menentukan karakteristik dan ciri wajah. Sementara itu, Metode Euclidean Distance digunakan sebagai metode pencocokan fitur wajah melalui citra hasil grayscale dan ekstraksi PCA sehingga dapat dilakukan pencocokkan fitur-fitur untuk dilanjutkan sebagai pengenalan wajah.

VI. UCAPAN TERIMA KASIH

Penghargaan tinggi sudah semestinya diberikan kepada segenap Lembaga Penelitian Universitas Pembangunan Pancabudi Medan (LPPM UNPAB), Rektor Universitas Pembangunan Pancabudi Medan, Fakultas Sains dan Teknologi, Program Studi Sistem Komputer, Atas dukungan terhadap karya penelitian ini dan fasilitas yang telah diberikan.

DAFTAR PUSTAKA

- Borkar, N.R. & Kuwelkar, S. 2017. *Real-Time Implementation Of Face Recognition System*. International Conference on Computing Methodologies and Communication (ICCMC). pp. 249-255.
- Elmahmudi, A. & ugail, H. 2019. *Deep Face Recognition using Imperfect Facial Data*. Future Generation of Computer Systems 99 pp. 213-225.
- Eleyan, A. & Demirel, H. 2005. *Face Recognition System based on PCA and Feedforward Neural Networks*. Proceedings of Computational Intelligence and Bioinspired Systems, pp. 935-942, ISBN: 978-3-540-26208-4, Spain, June 2005, Springer-Verlag, Barcelona.
- Eleyan, A. & Demirel, H. 2006. *PCA and LDA Based Face Recognition Using Feedforward Neural Network Classifier*. Proceedings of Multimedia Content Representation, Classification and Security, pp. 199-206, 978-3-540-39392-4, Turkey, September 2006, Springer-Verlag, Istanbul.
- James, C. & Nettikadan, D. 2019. *Student Monitoring System for School Bus using Facial Recognition*. The 3rd International Conference on Trends in Electronics and Informatics (ICOEI). pp. 659-663
- Jolliffe, I.T. 2002. *Principal Component Analysis*. 2nd Edition. Springer: New York.
- Johnson, W.A. & Wichern, D.W. 2007. *Applied Multivariate Statistical Analysis*. 6th Edition. Pearson Prentice Hall: New Jersey.
- Kirby, M. & Sirovich, L. 1990. *Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12 No. 1, (January 1990) pp.103-108, 0162-8828.
- Madhuram, M., et al. 2018. *Face Detection and Recognition using OpenCV*. International Research Journal of Engineering and Technology (IRJET), Vol.5 No.10. pp. 474-477
- Marijeta Slavkovic and Dubravka Jevtic. 2012. *Face Recognition Using Eigenface Approach*. Serbian Journal of Electrical Engineering, Vol. 9, No. 1, 2012.
- Nasution, MZ, et al. 2018. *PCA Based Feature Reduction to Improve the Accuracy of Decision Tree C4.5 Classification*. International Conference on Computing and Applied Informatics 2017. DOI: 10.1088/1742-6596/978/1/012058.
- Primandari, P.N. & Hardiansyah. 2018. *Ekstraksi Fitur Menggunakan Principal Component Analysis (PCA)*. Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat. Vol. 1 No. 1. pp. 66-74.
- Kotu, V. & Deshpande, B. 2015. *Predictive Analytics and Data Mining*. Morgan Kaufmann Publisher: San Francisco.
- Kresimir Delac and Mislav Grigic, "Preface, Face Recognition", 2007.
- Ramadhani, A.L., Musa, P. & Wibowo. E.P. 2017. *Human Face Recognition Application using PCA and Eigenface Approach*. The 2nd International Conference on Informatics and Computing. DOI:10.1109/IAC.2017.8280652.