

# PENGENALAN SESEORANG MENGGUNAKAN CITRA GARIS TANGAN

Bagus Fadzerie Robby<sup>1)</sup>, Resty Wulanningrum<sup>2)</sup>

<sup>1), 2)</sup> Universitas Nusantara PGRI Kediri  
<sup>1), 2)</sup> Jl. KH. Achmad Dahlan 76, Kediri, Jawa Timur 64112  
Email : [fadzerie@gmail.com](mailto:fadzerie@gmail.com)<sup>1)</sup>, [resty0601@gmail.com](mailto:resty0601@gmail.com)<sup>2)</sup>

## Abstrak

Telapak tangan menjadi sangat menarik untuk dikembangkan sebagai biometrika karena memiliki ciri yang lebih banyak dibanding dengan sidik jari dan geometri tangan. Permukaan area telapak tangan yang lebih luas dibandingkan sidik jari diharapkan dapat menghasilkan ciri yang memiliki kemampuan membedakan yang lebih dapat diandalkan. Hasil dari penelitian untuk identifikasi telapak tangan dengan menggunakan proses *preprocessing grayscale* dan deteksi tepi operator Sobel, dan *Principal Component Analysis (PCA)* dengan menggunakan sampel sebanyak 12 orang, dengan tiap orangnya sebanyak 10 sample citra telapak tangan. Hasil akurasi terbaik dari 5 skenario adalah pada skenario pertama, yaitu sebanyak 70 data training dan 30 data testing, hasil akurasi sebesar 93,3%.

**Kata kunci:** Telapak tangan, PCA, Sobel

## 1. Pendahuluan

Teknologi biometrika adalah suatu cara untuk mengenali seseorang berdasarkan karakteristik fisiologis (fisik) atau non-fisiologis (perilakunya). Karakteristik fisiologis terdiri dari elemen biometrika fisik bawaan antara lain garis-garis (rajah) pada telapak tangan, sidik jari, retina dan telapak tangan. Sedangkan karakteristik non-fisiologi adalah elemen biometrika yang muncul dari suatu proses belajar atau kebiasaan (tingkah laku).

Untuk sistem pengenalan dapat dikelompokkan menjadi 3, yaitu *Possessions Based*, *Knowledge Based* dan *Biometrics Based*. *Possessions Based* adalah sistem pengenalan yang berdasarkan suatu benda atau barang, seperti kunci dan kartu. Sedangkan *Knowledge Based* adalah sistem pengenalan yang berdasarkan pada sesuatu yang telah diketahui, seperti identitas pengguna (*userid*), *password*, dan nomer PIN. Dan yang terakhir adalah *Biometrics Based* yaitu sistem pengenalan yang menggunakan pendekatan berdasarkan biometrika. Kelemahan dari *Possessions Based* yaitu dapat hilang atau dicuri, dapat digunakan secara bersama-sama, dan mudah diduplikasi. Seperti *Possessions Based*, *Knowledge Based* juga memiliki beberapa kelemahan antara lain yaitu: lupa *password*, dapat digunakan secara bersama-sama, dan beberapa *password* mudah diperkirakan. Beda halnya dengan teknologi biometrika yang memiliki beberapa kelebihan yaitu *user* atau

pengguna tidak perlu menghafalkan *password*, tidak bisa di curi atau hilang dan tidak bisa di pertukarkan.

Pada telapak tangan mempunyai ciri atau karakteristik yang unik. Telapak tangan juga mempunyai suatu ciri geometri seperti bentuk, ukuran telapak tangan, lebar, dan panjang jari. Namun ciri geometri pada telapak tangan mempunyai beberapa kekurangan diantaranya, kurangnya keunikan dari ciri geometri sehingga ciri geometri ini mudah di palsu, kestabilan pada ciri geometri juga masih tergolong rendah, karena ciri geometri sepanjang tahun mungkin tidak sama yang di sebabkan oleh proses pertumbuhan pada manusia. Namun pada telapak tangan mempunyai suatu ciri yang bersifat stabil. Garis-garis (rajah) pada telapak tangan merupakan salah satu ciri yang bersifat stabil dan memiliki suatu keunggulan dan kelebihan di antaranya yaitu sulit untuk di palsu dan bersifat stabil karena sedikit mengalami perubahan dalam kurun waktu yang sangat lama.

Pada penelitian ini, proses identifikasi garis-garis (rajah) pada telapak tangan terlebih dahulu mengalami suatu *preprocessing image* yang bertujuan untuk mendapatkan ROI yaitu proses lokalisasi kemudian menghasilkan sub-*image*. Setelah didapatkan hasil dari *preprocessing image* maka tahap adalah pencarian jarak terdekatnya dengan menggunakan *Euclidean Distance*.

## 2. Pembahasan

### 2.1 Pengenalan Pola

Pada dasarnya sistem pengenalan pola terdiri atas tiga tahapan, yaitu: pemrosesan awal (*preprocessing*), ekstraksi fitur dan klasifikasi [1]. Pemrosesan awal antara lain dilakukan proses pengambilan sebagian area dari sebuah objek (*cropping*) untuk mendapatkan objek yang unik yang sering disebut dengan istilah *Region of Interest (ROI)*, menghilangkan *noise*, dan normalisasi pola. Ekstraksi fitur bertujuan untuk mereduksi data dengan melakukan pengukuran atas fitur- fitur atau properti tertentu.

Ekstraksi fitur akan melakukan karakteristik sebuah obyek untuk dikenali. Fitur- fitur sebuah obyek pada kategori yang sama mempunyai nilai yang hamper mirip (*similar*). Tahap selanjutnya adalah klasifikasi, bertujuan untuk menempatkan sebuah objek ke dalam sebuah kategori atau kelas. Tahapan pengenalan pola dapat digambarkan sebagai berikut:



Gambar 1. Proses pengenalan Pola.

### 2.2 Gray Scale (Citra skala keabuan)

Citra skala keabuan member kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai- nilai lain diantara nilai minimum (biasanya = 0) dan nilai maksimumnya. Banyaknya kemungkinan nilai dan nilai maksimumnya bergantung pada jumlah bit yang digunakan. Contohnya skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah  $2^4 = 16$ , dan nilai maksimumnya adalah  $2^4-1= 15$ , sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah  $2^8 = 256$ , dan nilai maksimumnya adalah  $2^8-1=255$ [1]

Citra true color dapat dikonversi menjadi citra keabuan dengan operasi titik. Secara mudahnya, intensitas didefinisikan sebagai nilai rerata dari ketiga nilai elemen warna, sehingga nilai keabuan yang merepresentasikan intensitas dapat dihitung dengan rumus:

$$k_o = \frac{Ri + Gi + Bi}{3}$$

Adapun hasil konversi *true color* ke *grayscale* ditunjukkan opeh gambar 2.



Gambar 2. Hasil konversi *true color* ke *grayscale*

### 2.3 Deteksi Tepi

Tepi merupakan salah satu fitur citra yang penting, karena dapat mewakili informasi yang penting dari objek dalam pemandangan. Informasi tepi

menghantarkan informasi tentang citra, yang berarti juga mengandung informasi tentang pemandangan. Tepi ditandai dengan adanya perubahan intensitas yang bersifat local di dalam citra dan dapat dilacak berdasarkan perubahan intensitas local ini. Tepi biasanya muncul pada batas antara dua daerah yang berbeda nilai intensitasnya dalam citra dan pelacakan tepi seringkali digunakan sebagai langkah awal dalam memperbaiki informasi dari citra. Deteksi tepi mempunyai kontribusi yang penting dalam pengembangan algoritma untuk analisis fitur- fitur citra, segmentasi citra, dan analisis pergerakan [2].

### 2.4 Operator Sobel

Sobel menempatkan penekanan atau pembobotan pada piksel- piksel yang lebih dekat dengan titik pusat jendela. Dengan demikian pengaruh piksel- piksel tetangga akan berbeda sesuai dengan letaknya terhadap gradien juga merupakan gabungan dari posisi mendatar dan posisi vertikal. Operator sobel adalah operator yang paling banyak digunakan sebagai pelacak tepi, karena kesederhanaan dan keampuhannya [2].

Susunan dari piksel- piksel di sekitar piksel  $p_o=(x,y)$  seperti ditunjukkan pada gambar 3.

$P_1$	$P_2$	$P_3$
$P_8$	$(x,y)$	$P_4$
$P_7$	$P_6$	$P_5$

Gambar 3. Susunan piksel- piksel tetangga dari piksel  $p_o=(x,y)$

Maka berdasarkan susunan piksel- piksel tetangga tersebut, besaran gradien yang dihitung dengan operator sobel adalah

$$M = \sqrt{s_x^2 + s_y^2}$$

Dimana M adalah besaran gradien yang dihitung pada titik tengah jendela dan turunan parsial dihitung dengan:

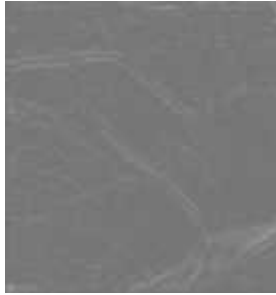
$$S_x = (p_3+cp_4+p_5) - (p_1+cp_8+p_7)$$

$$S_y = (p_1+cp_2+p_3) - (p_7+cp_6+p_5)$$

Dengan c suatu konstanta bernilai 2. Seperti dalam perhitungan gradien pada operator pelacak tepi lainnya,  $s_x$  dan  $s_y$  dapat diimplementasikan menggunakan jendela:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Hasil citra yang dilakukan deteksi tepi terlihat pada gambar 4 dengan menggunakan operator sobel.



Gambar 4. Hasil deteksi tepi dengan operator sobel

### 2.5 Principal Component Analysis

Principal Component Analysis (PCA) merupakan suatu perhitungan standar modern yang digunakan untuk analisis data pada macam-macam field atau multi dimensi sekumpulan data (dataset) khususnya pada bidang komputer grafik, karena kemudahannya dengan tidak membutuhkan parameter khusus dalam ekstraksi informasi yang berhubungan terhadap sekumpulan data yang meragukan [3]. Dengan usaha minimal PCA menyediakan alur bagaimana mengurangi kumpulan data yang kompleks ke dalam dimensi lebih kecil.

Algoritma ini diperkenalkan pada tahun 1933 oleh H. Hotelling, oleh karena itu sering juga disebut transformasi Hotelling. Untuk pertama kalinya PCA dikembangkan oleh para ahli statistik untuk mengeksplorasi hubungan sejumlah variable kualitatif yang dikembangkan oleh Karl Pearson pada tahun 1901. Karl Pearson membuat makalah yang menganalisis matrik korelasi yang berasal dari pengukuran tujuh variabel fisik untuk tiap orang dari 3000 kriminal [4]

### 2.6 Algoritma PCA

PCA merupakan teknik linear untuk memproyeksikan data vector yang berdimensi tinggi ke vector yang mempunyai dimensi lebih rendah.[5]

Jika jumlah data pelatihan adalah sebanyak m, maka dimensi vektornya adalah mxn. Apabila vector citra pelatihan mempunyai dimensi mxn tersebut diortogonalisasi dengan menentukan *eigenvector* dan *eigenvalue*, maka dimensinya akan berubah menjadi mxm, dimana  $m \ll n$ . Pengurangan dimensi yang sangat signifikan ini akan sangat membantu untuk mempercepat waktu komputasinya dalam melakukan klasifikasi. Namun sebelum klasifikasi, perlu dilakukan penyeleksian *eigenface*. [6]

Menurut Mauridhi Hery Purnomo, dari sebuah dataset citra pelatihan misalkan terdapat m citra pelatihan  $x = [x_1, x_2, x_3, \dots, x_m]$  yang ukuran dimensinya diasumsikan sebagai  $n = (w \cdot h)$  maka citra tersebut dapat dibentuk menjadi vektor kolom dengan ukuran  $(n \cdot 1)$  sebagai berikut:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \dots\dots 1$$

Keterangan:  $x$  = citra pelatihan  
 $x = [x_1, x_2, x_3, \dots, x_m]$

Dimana tiap nilai dari  $x_{nm}$  merupakan intensitas pixel pada suatu indeks *pixel* citra ke-n dari citra pelatihan ke-m. selanjutnya vektor kolom tersebut dilakukan normalisasi dengan menggunakan rumus :

$$X = \frac{X_k}{\sqrt{\sum_{k=1}^n X_k^2}} \dots\dots\dots 2$$

Keterangan:  $x_k$  : vektor kolom dari matriks x  
 n : jumlah dimensi

Dimana  $X_k$  adalah vektor kolom dari matriks X berordo  $(n \cdot m)$  dan N adalah dimensi dari ruang citra.

$$x = \begin{pmatrix} \frac{x_{11}}{\sqrt{x_{11}^2 + x_{12}^2 + \dots + x_{1n}^2}} & \frac{x_{12}}{\sqrt{x_{12}^2 + x_{13}^2 + \dots + x_{1n}^2}} & \dots & \frac{x_{1m}}{\sqrt{x_{1m}^2 + x_{1n}^2 + \dots + x_{1n}^2}} \\ \frac{x_{21}}{\sqrt{x_{21}^2 + x_{22}^2 + \dots + x_{2n}^2}} & \dots & \dots & \frac{x_{2m}}{\sqrt{x_{2m}^2 + x_{2n}^2 + \dots + x_{2n}^2}} \\ \dots & \dots & \dots & \dots \\ \frac{x_{n1}}{\sqrt{x_{n1}^2 + x_{n2}^2 + \dots + x_{nn}^2}} & \frac{x_{n2}}{\sqrt{x_{n2}^2 + x_{n3}^2 + \dots + x_{nn}^2}} & \dots & \frac{x_{nm}}{\sqrt{x_{nm}^2 + x_{nn}^2 + \dots + x_{nn}^2}} \end{pmatrix} \dots\dots 3$$

Hasil dari perhitungan matrik diatas adalah sebagai berikut:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & \dots & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \dots\dots 4$$

Setelah dilakukan normalisasi, matrik dari hasil perhitungan tersebut dilanjutkan dengan menghitung rata-rata matrik untuk mendapatkan matrik center dengan menggunakan rumus dibawah ini:

$$\mu = \frac{(x_{1,1} + x_{2,1} + x_{3,1} + \dots + x_{m,1})}{m} \dots\dots\dots 5$$

$$\mu_i = \sum_{j=1}^m X_{j,i} \dots\dots\dots 6$$

$$\mu_i = [\mu_1, \mu_2, \dots, \mu_n] \dots\dots\dots 7$$

keterangan :  $\mu$  : rata- rata matrik

dimana M adalah jumlah citra pelatihan. Rumus diatas dapat dijabarkan sebagai berikut

$$\mu = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{m1} & \mu_{m2} & \dots & \mu_{mn} \end{bmatrix} \dots\dots\dots 8$$

Keterangan:  $\mu$  = rata-rata matrik

Dan nilai dari  $[\mu_{i1} \mu_{i2} \mu_{i3} \dots \mu_{in}]$  pada baris ke  $i =$  nilai baris ke  $i+1$  dan berlaku  $\forall j, j \in 1,2,3, \dots m-1$ , maka persamaan (8) dapat digunakan untuk menghitung *zero mean*. *Zero mean* data dapat dimodelkan menggunakan persamaan

$$j,i = X_{j,i} - \mu \dots\dots\dots 9$$

Keterangan:  $\mu$  : zero mean  
 $\mu$  : rata-rata matrik

Atau dapat dimodelkan menggunakan matrik seperti persamaan berikut:

$$= \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} - \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{m1} & \mu_{m2} & \dots & \mu_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} x_{11} - \mu_{11} & x_{12} - \mu_{12} & \dots & x_{1n} - \mu_{1n} \\ x_{21} - \mu_{21} & x_{22} - \mu_{22} & \dots & x_{2n} - \mu_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} - \mu_{m1} & x_{m2} - \mu_{m2} & \dots & x_{mn} - \mu_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2n} \\ \dots & \dots & \dots & \dots \\ \phi_{m1} & \phi_{m2} & \dots & \phi_{mn} \end{bmatrix} \dots\dots\dots 10$$

Keterangan :  $\phi$  : zero mean  
 $\mu_{mn}$  : zero mean pada matrik mxn  
 $\mu$  : rata-rata matrik

Dari hasil perhitungan zero mean digunakan untuk mendapatkan nilai matrik kovarian selain dapat dimodelkan juga dilakukan dengan mengalihkan transpos zero mean, seperti terlihat pada persamaan di bawah ini:

$$C = \frac{1}{m-1} \sum_{j,i} x_{j,i} x_{j,i}^T$$

$$C = \frac{1}{m-1} (x_{j,i} - \mu_j) (x_{j,i} - \mu_j)^T \dots\dots\dots 11$$

Keterangan :  
 C : matrik kovarian  
 x : matrik awal  
 $\mu$  : rata-rata matrik  
 $\phi$  : zero mean

Persamaan (11) dapat diuraikan menggunakan matrik seperti pada persamaan

$$C = \frac{1}{m-1} \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2n} \\ \dots & \dots & \dots & \dots \\ \phi_{m1} & \phi_{m2} & \dots & \phi_{mn} \end{bmatrix} \times \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2n} \\ \dots & \dots & \dots & \dots \\ \phi_{m1} & \phi_{m2} & \dots & \phi_{mn} \end{bmatrix}^T$$

$$C = \frac{1}{m-1} \begin{bmatrix} c(x_1, x_1) & c(x_1, x_2) & \dots & c(x_1, x_m) \\ c(x_2, x_1) & c(x_2, x_2) & \dots & c(x_2, x_m) \\ \dots & \dots & \dots & \dots \\ c(x_m, x_1) & c(x_m, x_2) & \dots & c(x_m, x_m) \end{bmatrix} \dots\dots 12$$

Keterangan :  
 C : matrik kovarian  
 x : matrik awal  
 $\mu$  : rata-rata matrik  
 $\phi$  : zero mean

Berdasarkan persamaan (12) maka dimensi dari kovarian menjadi mxm, dengan berkurangnya jumlah dimensi yang sangat signifikan pada matrik kovarian, maka akan menyebabkan berkurangnya dimensi pada saat menentukan eigenvector sebagai vektor ciri. Untuk mendapatkan ciri dari suatu sampel yang direpresentasikan dalam bentuk matrik, maka dihitung

eigenvector dan eigenvalue dari matrik kovarian. Jika C adalah matrik bujur sangkar dengan ukuran sembarang m>1, maka vektor tak nol  $\Lambda$  pada  $R^n$  disebut eigenvector dari C jika  $C\Lambda$  suatu penggandaan skalar dari  $\Lambda$ , yang dihitung menggunakan persamaan (13)

$$C \Lambda = \Lambda \dots\dots\dots 13$$

Keterangan :  
 C : matrik kovarian  
 $\Lambda$  : eigenvector  
 $\lambda$  : eigenvalue

Skalar disebut sebagai eigenvalue dari C dan  $\Lambda$  disebut sebagai eigenvector dari C yang berpadanan terhadap  $\lambda$ . Untuk mendapatkan eigenvector dan eigenvalue, maka dari persamaan (13) dapat dituliskan menjadi persamaan

$$C \Lambda = \lambda \Lambda$$

$$(C - \lambda I) \Lambda = 0 \dots\dots\dots 14$$

$$\text{Det}(C - \lambda I) = 0$$

Keterangan :  
 C : matrik kovarian  
 $\Lambda$  : eigenvector  
 $\lambda$  : eigenvalue

Atau dituliskan menggunakan persamaan

$$\begin{vmatrix} \lambda & 0 & \dots & 0 \\ 0 & \lambda & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda \end{vmatrix} - \begin{vmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \dots & \dots & \dots & \dots \\ C_{m1} & C_{m2} & \dots & C_{mn} \end{vmatrix} = 0$$

$$\begin{vmatrix} \lambda - C_{11} & -C_{12} & \dots & -C_{1n} \\ -C_{21} & \lambda - C_{22} & \dots & -C_{2n} \\ \dots & \dots & \dots & \dots \\ -C_{m1} & -C_{m2} & \dots & \lambda - C_{mn} \end{vmatrix} = 0 \dots\dots 15$$

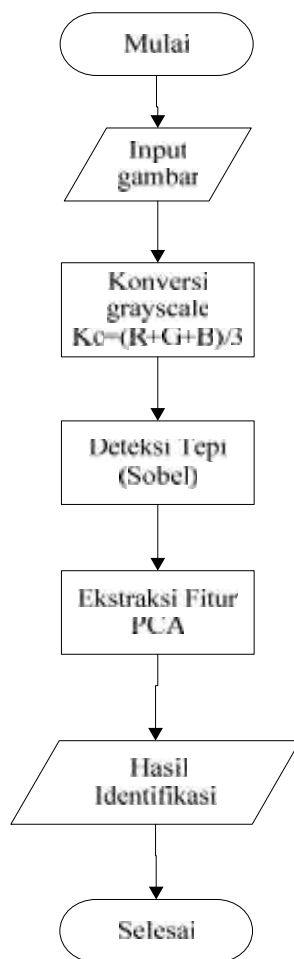
Keterangan :  
 C : matrik kovarian  
 $\Lambda$  : eigenvector  
 $\lambda$  : eigenvalue

Hasil dari persamaan (12) adalah berupa vektor, yaitu eigenvalue ( $\lambda$ ) diurutkan secara menurun dari nilai paling besar menuju nilai paling kecil ( $\lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_m$ ). Eigenvector ( $\Lambda$ ) yang bersesuaian dengan nilai terbesar dari eigenvalue mempunyai ciri yang paling dominan, sedangkan nilai eigenvector yang bersesuaian dengan eigenvalue yang paling kecil mempunyai ciri paling tidak dominan [6].

3 Analisa dan Rancangan sistem

Perancangan sistem diperlukan untuk mendapatkan alur yang sesuai dengan penelitian. Pada saat proses training akan ada 12 orang yang diambil citra telapak tangannya, tiap orang diambil sejumlah 10

sampel yang akan digunakan untuk proses *training* dan juga proses *testing*. Adapun perancangan secara garis besar akan ditunjukkan oleh gambar 5.



Gambar 5. Alur proses identifikasi

#### 4 Analisa hasil

Pengujian sistem identifikasi pada penelitian ini menggunakan sekitar 120 citra telapak tangan milik 10 orang dengan 10 sampel setiap orang. Dari data sampel ini akan dibuat skenario ujicoba, yaitu ditunjukkan oleh tabel 1.

Tabel 1. Skenario ujicoba

No.	Skenario	Data <i>training</i>	Data <i>Testing</i>
1.	Skenario 1	7	3
2.	Skenario 2	6	4
3.	Skenario 3	5	5
4.	Skenario 4	4	6
5.	Skenario 5	3	7

Skenario di atas dibuat dengan tujuan untuk mendapatkan hasil yang terbaik. Setiap skenario akan mempunyai nilai akurasi yang berbeda-beda. Nilai –

nilai itu yang akan digunakan untuk menganalisa, pada skenario mana yang hasilnya terbaik.

Dari hasil skenario yang diibuat didapatkan hasil seperti ditunjukkan oleh tabel 2.

Tabel 2. Hasil ujicoba

No.	Data <i>training</i>	Data <i>testing</i>	Akurasi %
1.	70	30	93.3
2.	60	40	90
3.	50	50	88
4.	40	60	86.6
5.	30	70	85.7

Pada tabel 2, ujicoba dilakukan tanpa adanya proses reduksi dimensi. Kemudian dilakukan ujicoba lagi dengan melakukan reduksi dimensi sebesar 10, 20 dan 30. Analisa ujicoba pada skenario 1 ditunjukkan oleh tabel 3.

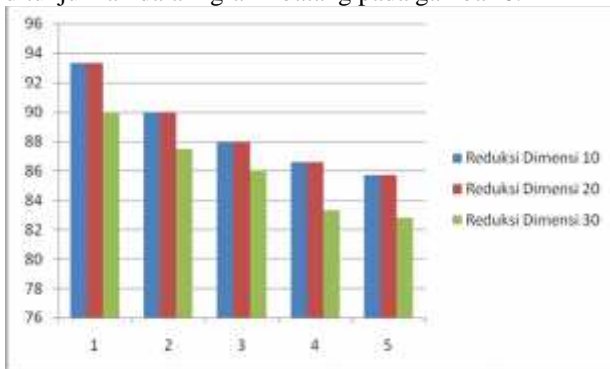
Tabel 3. Hasil reduksi dimensi pada skenario

Skenario 1			
Reduksi	Data <i>Training</i>	Data <i>Testing</i>	Akurasi %
10	70	30	93.3
20	70	30	93.3
30	70	30	90
Skenario 2			
Reduksi	Data <i>Training</i>	Data <i>Testing</i>	Akurasi %
10	60	40	90
20	60	40	90
30	60	40	87.5
Skenario 3			
Reduksi	Data <i>Training</i>	Data <i>Testing</i>	Akurasi %
10	50	50	88
20	50	50	88
30	50	50	86
Skenario 4			
Reduksi	Data <i>Training</i>	Data <i>Testing</i>	Akurasi %
10	40	60	86.6
20	40	60	86.6
30	40	60	83.3
Skenario 5			
Reduksi	Data <i>Training</i>	Data <i>Testing</i>	Akurasi %
10	30	70	85.7
20	30	70	85.7
30	30	70	82.8

Dari hasil ujicoba yang dilakukan berdasarkan tabel 3, maka hasil yang terbaik yaitu pada skenario ujicoba 1 yaitu sebesar 93.3% dengan reduksi dimensi sebesar 10 dan 20. Hal ini dipengaruhi oleh banyaknya jumlah data *training* dengan besar reduksi dimensi. Semakin besar data *training*, maka hasilnya akan semakin bagus akurasinya, semakin besar nilai reduksi dimensi, maka hasilnya pun semakin menurun akurasinya.

Hasil perbandingan dari 5 skenario yang dibuat dapat dilihat pada gambar 6. Yaitu grafik hubungan

anantara skenario ujicoba dengan hasil akurasi akan ditunjukkan dalam grafik batang pada gambar 6.



Gambar 6. Grafik perbandingan

Dari gambar 6 dapat dilihat bahwa hasil skenario ujicoba dengan menggunakan reduksi dimensi sebesar 10 dan 20 hasil akurasi adalah sama. Yaitu sebesar 93.3 %.

### 3. Kesimpulan

Hasil dari penelitian untuk identifikasi telapak tangan dengan menggunakan proses preprosesing grayscale dan deteksi tepi operator sobel, dan *Principal Component Analysis (PCA)* dengan menggunakan sampel sebanyak 12 orang, dengan tiap orangnya sebanyak 10 sample citra telapak tangan. Hasil akurasi terbaik dari 5 skenario adalah pada skenario pertama, yaitu sebanyak 70 data *training* dan 30 data *testing*, hasil akurasi sebesar 93,3%.

Hasil akurasi terbaik ini didapatkan dengan melakukan beberapa skenario ujicoba. Semakin besar data yang digunakan pada saat *training*, maka akan semakin besar pula hasil akurasi. Dan semakin besar reduksi dimensinya, maka hasilnya semakin menurun, dikarenakan hilangnya informasi akibat proses reduksi dimensi.

### Daftar Pustaka

- [1] Achmad, Balza dan Firdausy, Kartika..”*Teknik Pengolahan Citra Digital Menggunakan Delphi*”. Ardi Publisng. Yogyakarta. 2005
- [2] Ahmad Usman.”*Pengolahan Citra Digital dan Teknik Pemrogramannya*”. Graha Ilmu. Yogyakarta. 2005.
- [3] Smith, Lindsay. “*A Tutorial on Principal Component Analysis*”. 2002.Sumber [http://www.cs.otago.ac.nz/cosc453/student\\_tutorials/principal\\_components.pdf](http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf). Diakses pada 19 Juni 2013.
- [4] Leon, Steven J.” *Aljabar Linear dan Aplikasinya*”. Erlangga, Jakarta. 2001.
- [5] Turk,M and Pentlant. “*Face Recognition Using Eigenfaces*” Proc. IEEE Conf. of Computer Vision and Pattern Recognition, 13, Pp 586-591. 1991.
- [6] Purnomo, M.H, Arif Muntasa, “*Konsep Pengolahana Citra Digital dan Ekstraksi Fitur*”. Yogyakarta: Graha Ilmu. 2010.
- [7] Gonzalez, R.C. and Woods, R.E, *Digital Image Processing*, Prentice-Hall,inc. New Jersey.2002.

### Biodata Penulis

**Bagus Fadzerie Robby**, memperoleh gelar Sarjana Komputer (S.Kom), jurusan Teknik Informatika Universitas Trunojoyo Madura, lulus pada tahun ..... Saat ini sebagai kepala Puskom dan Staf Pengajar Program Sarjana Teknik Informatika UNP Kediri.

**Resty Wulanningrum**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika Universitas Trunojoyo Madura, lulus pada tahun 2011. Saat ini sebagai Staf Pengajar Program Sarjana Teknik Informatika UNP Kediri.