

# Pengenalan Wajah dengan Metode *Principal Component Analysis* dan Jaringan Syaraf Tiruan Berbasis Android

## FACE RECOGNITION USING *PRICIPAL COMPONENT ANALYSIS* AND NEURAL NETWORK BASED ON ANDROID

Kevin Alamsyah Yuwono<sup>1</sup>, Iwan Iwut Tritoasmoro, S.T, M.T.<sup>2</sup>, Irma Safitri, S.T., M.T.<sup>3</sup>

<sup>1,2</sup> Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup> kevinalamsyah @student.telkomuniversity.ac.id, <sup>2</sup>iwaniwut@telkomuniversity.ac.id

<sup>3</sup>irmasafirti@telkomuniversity.ac.id

### Abstrak

Berbagai metode telah dikembangkan untuk mencapai tingkat akurasi yang tinggi dan untuk menciptakan sistem pengenalan wajah yang handal. Diantaranya yang paling terkenal adalah *Principle Component Analysis (PCA)*. Walaupun proyeksi PCA cukup optimal untuk reduksi dimensi namun ternyata PCA kurang optimal dalam pemisahan kelas. Maka dari itu, PCA bisa dikombinasikan dengan metode lain yang lebih baik dalam pemisahan kelas. Salah satu contohnya adalah metode Jaringan Syaraf Tiruan (JST).

Pada tugas akhir ini, membangun sebuah sistem pengenalan wajah dengan menggunakan metode *Principal Component Analysis (PCA)* untuk ekstraksi dan memanfaatkan metode Jaringan Syaraf Tiruan (JST) sebagai klasifikasi. Sistem yang dibangun akan mampu mengenali identitas dari wajah masukan berdasar *database* yang telah disimpan sebelumnya. Output dari sistem ini berupa aplikasi berbasis android.

Dari hasil penelitian, didapatkan kesimpulan bahwa parameter yang paling ideal untuk penelitian ini menggunakan nilai *threshold* 0,3, jumlah *hidden layer* 5, dan metode fungsi pelatihan *traincgp* dengan menghasilkan tingkat akurasi dan waktu komputasi yang paling baik. Berdasarkan parameter-parameter tersebut didapatkan tingkat akurasi terbesar adalah 94% dengan waktu komputasi yang dibutuhkan adalah 0,47292 detik.

**Kata Kunci:** Pengenalan wajah, PCA, JST, Android

### Abstract

Various methods have been developed to achieve a high level of accuracy and to create a reliable face recognition system. Among the most famous are the Principle Component Analysis (PCA). Although PCA projections are optimal for dimensional reduction, PCA is less optimal in class separation. Therefore, PCA can be combined with other methods that are better in class separation. One example is the method of Artificial Neural Networks (ANN).

In this final project, build a face recognition system using the Principal Component Analysis (PCA) method for extraction and utilize the method of Artificial Neural Networks (ANN) as a classification. The system built will be able to recognize the identity of the input face based on the previously stored database. The output of this system is an Android-based application..

From the results of the study, it was concluded that the most ideal parameters for this study used a threshold value of 0.3, the number of layer 5 hidden, and the *traincgp* training function method by producing the best level of accuracy and computation time. Based on these parameters the greatest accuracy is obtained 94% with the computation time needed is 0,47292 seconds.

**Keywords :** Face recognition, PCA, Neural Network, Android

### 1. Pendahuluan

Identifikasi adalah penentuan atau pemastian identitas orang berdasarkan pada ciri khas yang terdapat pada orang tersebut. Pada dasarnya setiap manusia memiliki sesuatu yang unik atau ciri khas yang menjadikan keunikan tersebut sebagai identitas diri. Salah satu cara yang dapat dilakukan untuk mengidentifikasi identitas seseorang adalah dengan cara mengambil data karakteristik alami manusia, contohnya wajah.

Pengenalan wajah adalah salah satu teknik biometrik yang menggunakan wajah sebagai masukannya. Pengenalan wajah seseorang dapat dilakukan dengan adanya bantuan aplikasi. Salah satu contoh aplikasi sistem pengenalan wajah ini adalah *Matlab*. Pada penelitian pengenalan wajah digunakan berbagai metode metode yang mempengaruhi tingkat akurasi penelitian.

Metode yang paling sering digunakan dalam sistem pengenalan wajah adalah *Principal Component Analysis (PCA)*. Metode PCA adalah metode ekstraksi yang mampu mengidentifikasi ciri tertentu yang menjadi karakteristik seseorang dalam hal ini adalah wajah. Tetapi kelemahan pada metode PCA adalah kurang optimal dalam klasifikasi sehingga mempengaruhi tingkat akurasi identifikasi wajah [1].

Untuk mengatasi kelemahan di atas, metode PCA dapat dikombinasi dengan metode yang handal dalam klasifikasi contohnya Jaringan Syaraf Tiruan (JST). Metode JST merupakan salah satu metode klasifikasi yang meniru cara kerja otak manusia. Maka pada Tugas Akhir ini penulis merancang sebuah sistem pengenalan wajah dengan menggunakan metode *Principal Component Analysis* (PCA) sebagai ekstraksi cirinya dan menggunakan metode Jaringan Syaraf Tiruan (JST) sebagai klasifikasinya. Aplikasi diproses menggunakan matlab dengan *output* android.

## 2. Dasar Teori

### 2.1 Citra Digital

Citra dapat diartikan sebagai suatu fungsi kontinu dari intensitas cahaya  $(x,y)$  dalam bidang dua dimensi, dengan  $(x,y)$  menyatakan suatu koordinat spasial dan nilai  $f$  pada setiap titik  $(x,y)$  menyatakan intensitas atau tingkat kecerahan atau derajat keabuan (*brightness* atau *grey level*) dimana secara matematis dapat dirumuskan bahwa  $0 < f(x,y) < \infty$ . Citra digital mengandung sejumlah elemen-elemen dasar. Elemen-elemen dasar tersebut dapat dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam *computer vision*.

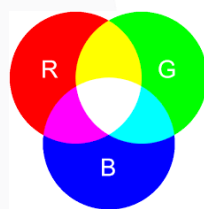
### 2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah pemrosesan citra dua dimensi atau tiga dimensi oleh komputer. Masukan dari sistem komputer tersebut berupa sinyal elektrik yang dapat dipresentasikan dalam bentuk numerik. Contohnya adalah kamera digital, *web camera*, dan lain-lain. Hasil dari proses tersebut harus dapat dinilai oleh mata manusia melalui suatu penampil (*display*), biasanya berupa *graphic monitor* [6].

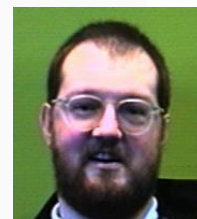
#### 2.2.1 Citra Red, Green, Blue (RGB)

Citra RGB adalah citra yang setiap pikselnya dibentuk oleh susunan campuran warna Merah, hijau dan biru yang merupakan warna primer. Setiap warna dasar diberi rentang nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini akan diperoleh warna campuran sebanyak  $256 \times 256 \times 256 = 1677726$  jenis warna[4].

(a) Komposisi warna RGB



(b) Citra sapi RGB



Gambar 1 (a) Komposisi warna RGB (b) Citra sapi RGB

#### 2.2.2 Citra Biner

Citra biner adalah citra dimana piksel-pikselnya hanya memiliki dua buah nilai derajat keabuan (grayscale) yaitu hitam dan putih. *Pixel-pixel* (picture elements) suatu objek akan bernilai 1 sedangkan *pixel-pixel* latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi, pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam.



Gambar 2 Citra Biner

### 2.3 Principal Component Analysis

*Principal Component Analysis* (PCA) secara sederhana adalah transformasi linear untuk menentukan sistem koordinat yang baru dari sebuah dataset. PCA sendiri pada dasarnya bertujuan untuk menyederhanakan informasi dengan cara mereduksi informasi data yang besar dari sebuah

citra wajah tanpa menghilangkan informasi yang ada pada citra wajah tersebut. Dengan reduksi ini maka waktu komputasi dapat dikurangi dan kompleksitas dari citra wajah yang tidak perlu dapat dihilangkan.

Berdasarkan [2][4][6] penyusunan algoritma PCA terdiri dari :

#### 1. Normalisasi masukan

Maksudnya adalah mengambil ciri-ciri yang menonjol dari sebuah citra. Misalnya terdapat sebuah citra berukuran 64 x 64 piksel. Ubah citra tersebut menjadi sebuah matriks kolom dengan ukuran 4096 x 1. Jika terdapat n citra maka matriksnya menjadi 4096 x n. Berikut ilustrasinya yang ditunjukkan pada rumus 2.4 :

$$u = \begin{bmatrix} u(1,1) & u(1,2) & \cdots & u(1,n) \\ u(2,1) & u(2,2) & \cdots & u(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ u(4096,1) & u(4096,2) & \cdots & u(4096,n) \end{bmatrix} \quad (1)$$

Dari matriks tersebut kemudian dicari citra rata-ratanya dengan merata-ratakan setiap barisnya. Berikut rumus 2.5 menghitung nilai rata-rata :

$$\mu = \frac{1}{N} \sum_{k=1}^N x_k \quad (2)$$

#### 2. Setiap citra pada matriks citra awal dikurangi dengan citra rata-rata

$$\phi_i = u_i - \mu \quad (3)$$

Dimana  $i = 1 \dots N$

$N$  = Ukuran Citra

Simpan  $\phi$  dalam matriks A sehingga :

$$A = [\phi_1; \phi_2; \phi_3; \cdots; \phi_N] \quad (4)$$

#### 3. Mencari matriks kovarian

Tujuan pencarian matriks kovarian ini adalah untuk mempermudah pencarian nilai *eigen* dan *vector eigen*. Nilai dari matriks kovarian citradapat dicari dengan persamaan 2.8 sebagai berikut :

$$L = A^T A \quad (5)$$

#### 4. Mencari vektor eigen dan nilai eigen

Vektor eigen dan nilai eigen dapat diperoleh dari persamaan 2.9 berikut.

$$L V' = \lambda V' \quad (6)$$

Keterangan :

$L$  = Matriks kovarian

$V'$  = Vektor eigen

$\lambda$  = Nilai dari eigen L

Setelah diperoleh nilainya, kemudian diurutkan *eigen value* ( $\lambda$ ) dan *eigen vector* ( $V'$ ) dari besar ke kecil berdasarkan urutan nilai *eigen*. Kemudian hitung nilai matriks *eigenface* dengan persamaan 2.10 sebagai berikut :

$$Eigenface = A V \quad (7)$$

#### 5. Transformasi PCA

Hitung project image dari citra dengan persamaan 2.11 sebagai berikut :

$$Project\ image = Eigenface^T A \quad (8)$$

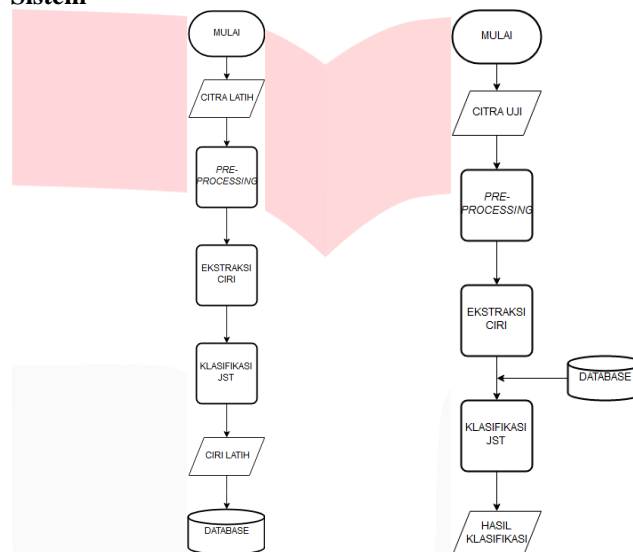
## 2.4 Jaringan Syaraf Tiruan

Jaringan saraf tiruan merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem saraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut neuron, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi.

Untuk membuat jaringan saraf tiruan, dibutuhkan parameter, seperti jumlah *hidden neuron*, *hidden layer*, *epoch*, *learning rate*, dan *momentum*. *Hidden neuron* adalah neuron yang tidak ada pada *input layer* maupun *output layer*, *hidden layer* adalah layer yang menghubungkan input dengan output, *epoch* adalah nilai yang menentukan kapan pelatihan akan berhenti, *learning rate* adalah parameter pengontrol perubahan bobot dan bobot bias pada saat pelatihan, sedangkan *momentum* digunakan untuk mencegah sistem konvergen pada kondisi minimum lokal [5].

## 3. Pembahasan

### 3.1 Diagram Alur Sistem



**Gambar 3** Diagram Alur Sistem

Bisa dilihat pada gambar 2, proses identifikasi dibagi menjadi dua proses, yaitu proses latih dan proses uji. Proses latih merupakan proses pencarian nilai parameter yang menjadi acuan untuk *database* program, dimana nilai parameter tersebut dicocokkan dengan citra uji untuk kelas yang nantinya akan dilakukan proses klasifikasi.

#### 3.1.1 Akuisisi Citra

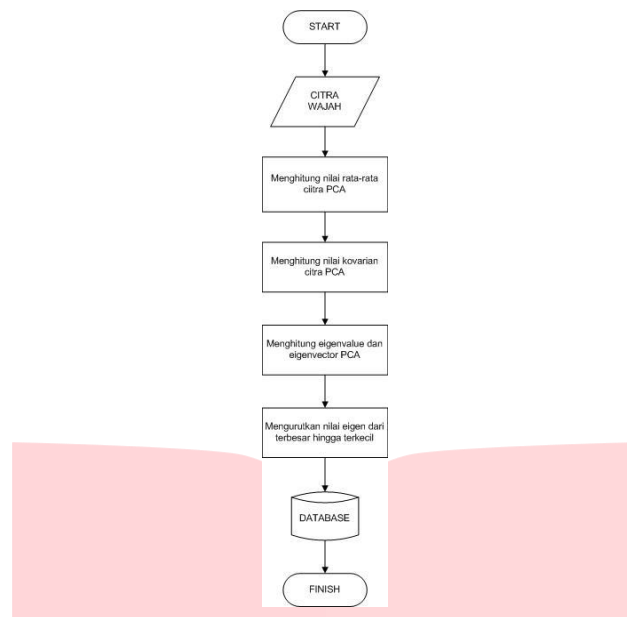
Citra yang didapatkan akan terbagi menjadi citra latih dan citra uji. Objek penelitian pada tugas akhir ini adalah wajah. Data citra wajah didapatkan dari *University of Essex*, Inggris [9] dengan format JPG.



**Gambar 4** Citra Wajah

#### 3.1.2 Ekstraksi Ciri

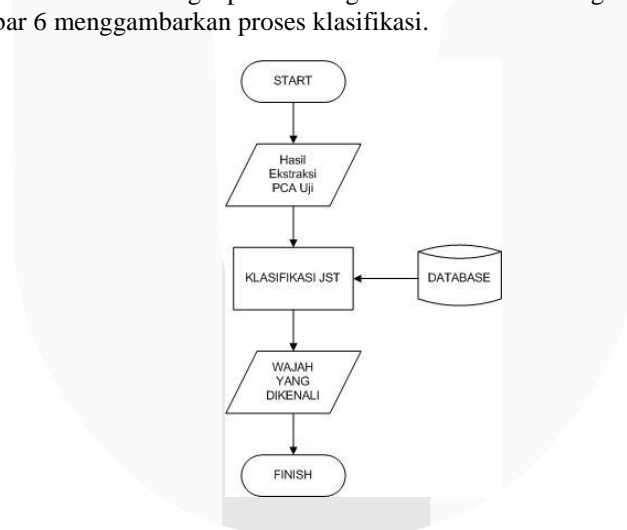
Pada tahap ini digunakan ekstraksi fitur *Principal Component Analysis*. Ekstraksi fitur ini menghasilkan fitur latih yang kemudian menjadi *database* dan fitur uji yang diujikan untuk menguji apakah wajah dikenali berdasarkan pada *database* dan tidak dikenali jika tidak terdapat pada *database*.



Gambar 5 Diagram ekstraksi ciri

3.1.3 Klasifikasi

Pada tahap ini dirancang klasifikasi dengan metode Jaringan syarat tiruan. Algoritma yang dipakai adalah algoritma *Backpropagation*. Pada tahap ini hasil dari citra yang telah melalui proses ekstraksi ciri akan diklasifikasi menurut kelasnya. Melalui *database* yang telah dimasukkan sebagai pembandingan akan dicocokkan dengan data yang diuji. Pada gambar 6 menggambarkan proses klasifikasi.



Gambar 6 Diagram alir klasifikasi

3.2 Performansi Sistem

Akurasi merupakan ukuran ketepatan sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang benar. Secara matematis dapat dituliskan sebagai berikut :

1. Akurasi sistem

Akurasi merupakan ukuran ketepatan sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang benar. Secara matematis dapat dituliskan sebagai berikut:

$$\text{Akurasi kelas} = \frac{\text{Jumlah data benar}}{\text{Jumlah data keseluruhan}} \times 100\% \tag{9}$$

2. Waktu komputasi

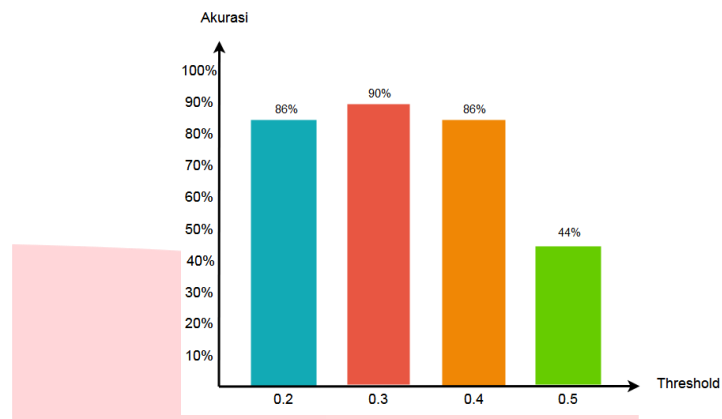
Waktu komputasi adalah waktu yang dibutuhkan sistem melakukan suatu proses. Pada sistem ini, waktu komputasi dihitung menggunakan waktu selesai dikurangi waktu mulai, sehingga akan didapatkan waktu komputasi sistem. Secara sistematis dapat dituliskan sebagai berikut :

$$\text{Waktu Komputasi} = \text{Waktu selesai} - \text{Waktu mulai} \tag{10}$$

### 3.3 Hasil Pengujian Sistem

#### 3.3.1 Hasil Pengujian Skenario 1

Skenario dari pengujian ini mengenai pengaruh perubahan nilai *threshold* dari citra yang diolah sistem terhadap akurasi sistem. Perubahan nilai *threshold* dari 0,2 sampai 0,5. Adapun hasil yang didapat mengenai pengujian parameter *threshold* ini tersaji pada gambar 6

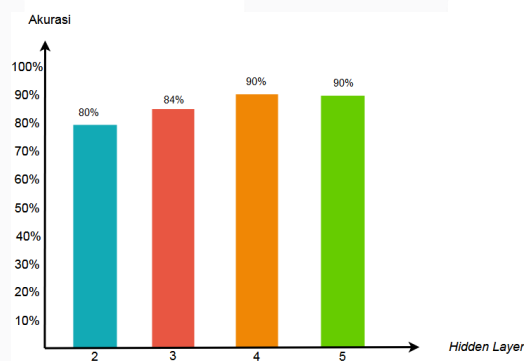


**Gambar 7** Grafik Pengaruh *Threshold* Terhadap Tingkat Akurasi

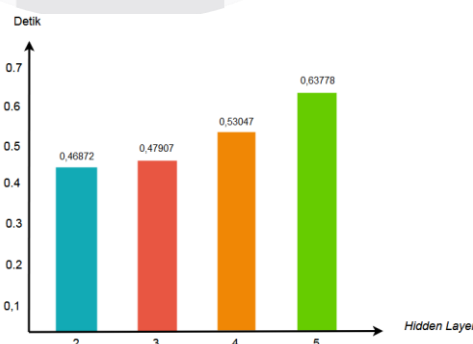
Pada gambar 7 membuktikan bahwa nilai *threshold* yang paling ideal sistem ini adalah 0,3 dengan menghasilkan tingkat akurasi yang lebih besar dari yang lainnya yaitu 90%.

#### 3.3.2 Hasil Pengujian Skenario 2

Pada skenario pengujian ini dilakukan analisis terhadap parameter klasifikasi jumlah *hidden layer*. Jumlah *hidden layer* yang digunakan adalah 2, 3, 4, dan 5. Adapun hasil yang didapatkan tersaji pada gambar 8 dan gambar 9.



**Gambar 8** Grafik Pengujian Jumlah *Hidden Layer* Terhadap Tingkat Akurasi

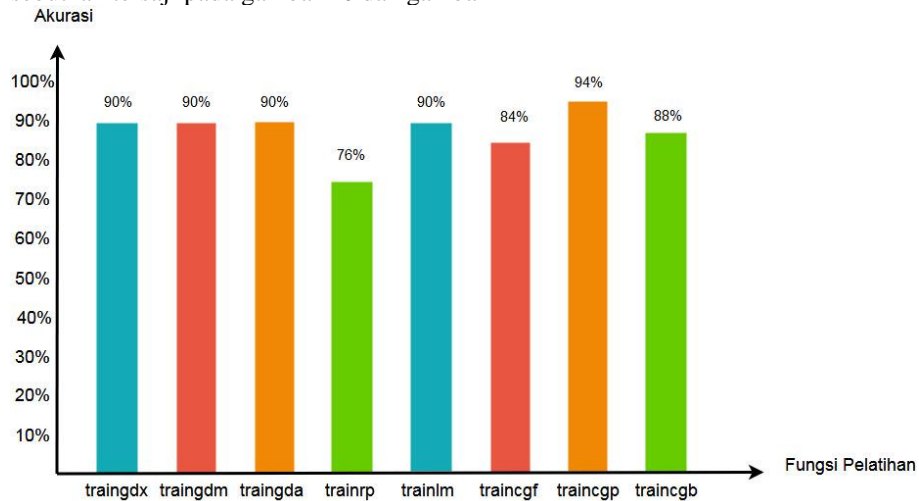


**Gambar 9** Grafik Pengujian Jumlah *Hidden Layer* Terhadap Waktu Komputasi

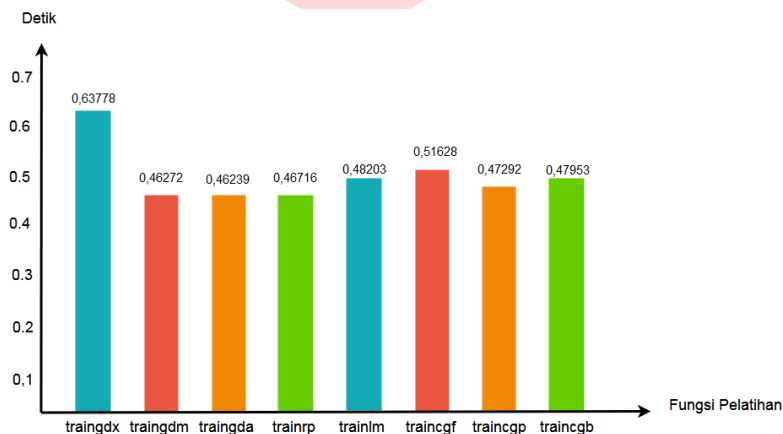
#### 3.3.3 Hasil Pengujian Skenario 3

Pada skenario pengujian ini dilakukan analisis parameter klasifikasi yang penggunaan

fungsi pelatihan pada metode JST. Pada metode JST memiliki beberapa fungsi pelatihan yaitu *traingdx*, *traingdm*, *traingda*, *trainrp*, *trainlm*, *traingcf*, *traingcp*, dan *traingcb*. Adapun hasil yang didapat mengenai pengujian berdasarkan fungsi-fungsi pelatihan yang telah disebutkan tersaji pada gambar 10 dan gambar 11



**Gambar 10** Grafik Pengaruh Penggunaan Fungsi Pelatihan Terhadap Tingkat Akurasi



**Gambar 11** Grafik Pengaruh Penggunaan Fungsi Pelatihan Terhadap Waktu Komputasi

Pada gambar 10 dan gambar 11 terlihat bahwa perbedaan waktu komputasi tidaklah signifikan. Perubahan fungsi pelatihan tidak terlalu berpengaruh terhadap perubahan waktu komputasi. Sementara terhadap tingkat akurasi sistem perubahan fungsi pelatihan cukup berpengaruh. Dalam percobaan ini didapat waktu komputasi 0,47292 detik dan nilai akurasi sebesar 94%. Nilai ini merupakan tingkat akurasi terbesar yang pernah tercapai selama percobaan dengan menggunakan fungsi pelatihan *traingcp*.

#### 4. Simpulan

Berdasarkan hasil analisis terhadap skenario pengujian yang telah dilakukan pada Tugas Akhir ini, maka dapat ditarik kesimpulan sebagai berikut :

1. Fungsi pelatihan yang digunakan mempengaruhi tingkat akurasi sistem. Tingkat akurasi tertinggi dicapai ketika fungsi pelatihan yang digunakan adalah *traingcp*. Sistem akan berhenti saat *epoch* sudah mencapai batas yang ditentukan atau saat MSE yang diinginkan tercapai. Semakin MSE mendekati 0 maka akurasi yang di hasilkan akan semakin baik.
2. Semakin banyak *hidden layer* semakin lama waktu komputasi yang dibutuhkan. Dikarenakan semakin banyaknya *hidden layer* semakin banyak pula nilai pembanding. Tingkat akurasi dan waktu komputasi terbaik tercapai ketika jumlah *hidden layer* adalah 4.

3. Nilai *threshold* juga berpengaruh terhadap tingkat akurasi sistem. Nilai yang ideal untuk digunakan dalam sistem ini adalah 0,3.
4. Penerapan metode *Principal Component Analysis* (PCA) dan klasifikasi Jaringan Syaraf Tiruan (JST) dalam aplikasi pendeteksi wajah mencapai tingkat akurasi terbaik sebesar 94% dan waktu komputasi 0,47292 detik. Tingkat akurasi tertinggi dicapai ketika fungsi pelatihan yang digunakan adalah *traincgp* dengan *hidden layer* 5 dan *threshold* 0,3.

#### Daftar Pustaka

- [1] Fadiansyah., Sari, J.Y., dan Ningrum, I.P. 2017. “Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan K Nearest Neighbor.” Fakultas Teknik Universitas Halu Oleo : Kendari.
- [2] Syakhala, Abdu., Puspitaningrum, Diyah., dan Purwandari, Endina. 2015. “Perbandingan Metode Principal Component Analysis (PCA) Dengan Hidden Markov Model (HMM) Dalam Pengenalan Identitas Seseorang Melalui Wajah.” Fakultas Teknik Informatika Universitas Bengkulu : Bengkulu.
- [3] Sepritahara. “Sistem Pengenalan Wajah (Face Recognition) Menggunakan Metode Hidden Markov Model (HMM).” Fakultas Teknik Elektro Universitas Indonesia : Jakarta.
- [4] Fitriyah, Nurani. 2016. “Implementasi dan Analisis Sistem Pengenalan Wajah Pada Video Pengawasan Untuk Ruang Terbuka Dengan Metode Fisherface.” Fakultas Teknik Elektro Telkom University : Bandung.
- [5] Dewangga, Dhion, A . 2014. “Desain dan Implementasi Steganalisis Citra Digital Menggunakan Metode Klasifikasi Jaringan Saraf Tiruan Pada Android Mini PC.” Fakultas Teknik Elektro Telkom University : Bandung.
- [6] Dharmawan, Willy. 2012. “Implementasi Prototype Pengenalan Wajah Manusia Berbasis Android Untuk Sistem Presensi.” Fakultas Teknik Elektro Telkom University : Bandung.
- [7] Akariman, Qawlan. 2015. “Implementasi dan Analisis Pengenalan Wajah Dengan Algoritma Local Binary Pattern Pada Kepala Robot Berbasis Android.” Fakultas Teknik Elektro Telkom University : Bandung
- [8] Prakarsa, Taufan. 2009. “Pengenalan Wajah Manusia Menggunakan Metode PCA (Principal Component Ananalysis) dan Jaringan Fungsi Basis Radial” Fakultas Teknik Elektro Telkom University : Bandung
- [9] University of Essex, UK. “Face Recognition Data” <https://cswww.essex.ac.uk/mv/allfaces/index.html> (diakses 24 November 2018)
- [10] Siang, Jong Jek. 2009. “Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB” Yohyakarta: ANDI OFFSET
- [11] <https://pemrogramanmatlab.com/2015/08/24/jaringan-syaraf-tiruan-untuk-identifikasi-wajah/#more-998> (diakses 22 November 2018)