

# Sistem Pengenalan Individu Berbasis Citra Wajah 3D dengan Jaringan Syaraf Tiruan

Muhammad Sindu Ramadhan<sup>1</sup>, Ledy Novamizanti<sup>2</sup>, Eko Susatio<sup>3</sup>

Program Studi Teknik Telekomunikasi,

Fakultas Teknik Elektro,

Universitas Telkom, Bandung

<sup>1</sup>sinduramadhan16@gmail.com, <sup>2</sup>ledyaldn@telkomuniversity.ac.id, <sup>3</sup>maharusdi@gmail.com

## Ringkasan

Pengenalan individu adalah sebuah cara yang banyak digunakan dalam bidang medis, teknologi, dan keamanan. Wajah adalah objek biometrika yang sering digunakan untuk pengenalan maupun sistem keamanan. Pada wajah terdapat beberapa karakteristik yang dapat digunakan dalam pengenalan, seperti ukuran mata, hidung dan bibir, jarak mata, hidung dan bibir, serta pola kedalaman bagian bagian wajah. Penelitian ini merancang sistem pengenalan individu dengan citra wajah individu dalam bentuk 3 dimensi. Citra wajah 3D diambil sebanyak 48 foto, di mana 48 foto tersebut merepresentasikan bentuk wajah secara keseluruhan. Dari 48 foto yang telah diambil akan dipartisi menjadi 3 partisi dan 6 partisi. Partisi tersebut akan digunakan sebagai masukan proses ekstraksi ciri. Metode yang digunakan untuk ekstraksi ciri adalah *Iterative Closest Point* (ICP) dan diklasifikasikan dengan metode Jaringan Syaraf Tiruan (JST). Penelitian ini menghasilkan performansi sebesar 78.57%. Sistem ini dapat menjadi salah satu alternatif pengenalan identitas seseorang dan diharapkan penelitian ini dapat menjadi referensi penelitian lanjut mengenai pengenalan individu berbasis citra 3D di masa mendatang.

**Kata kunci:** Wajah 3D, ICP, JST

## 1. Pendahuluan

Dewasa ini, kemajuan teknologi berkembang sangat pesat. Ada banyak dampak positif dan negatif yang didapat dari kemajuan teknologi. Kemajuan teknologi dimanfaatkan oleh sebagian orang untuk tindak kejahatan, antara lain, memanipulasi identitas seseorang, mengambil informasi penting yang dimiliki oleh seseorang. Oleh karena itu, identifikasi seseorang perlu digunakan agar keamanan privasi setiap orang terjaga. Identifikasi dengan menggunakan biometrika adalah salah satu cara yang banyak digunakan.

Sistem biometrika adalah suatu pengembangan metode dasar dari sistem identifikasi dengan menggunakan karakteristik dari manusia sebagai objeknya termasuk wajah, sidik jari, tanda tangan, telapak tangan, iris mata, telinga, suara dan DNA [1]. Sistem biometrika akan mencocokkan identitas seseorang dengan suatu data ciri yang sudah diambil melalui proses registrasi [2]. Wajah adalah objek yang sangat familiar digunakan untuk pengenalan, sebagai contoh: KTP, SIM dan *ID Card* lainnya yang memasukkan foto wajah seseorang dalam identitasnya.

Pengenalan wajah merupakan salah satu teknik identifikasi dalam biometrik yang menggunakan wajah individu terkait sebagai parameter utamanya [3]. Pengenalan individu berbasis wajah mengacu pada letak dan bentuk mata, hidung, bibir, alis dan dagu

atau dengan citra bagian wajah yang dapat mewakili wajah sebagai muka yang dikenali [4]. Namun, ada beberapa masalah dalam merencanakan sistem pengenalan wajah. Hal ini disebabkan variasi wajah manusia sangat beragam seperti kondisi pencahayaan, ekspresi, rotasi, sudut pengambilan gambar kamera, penuaan, *make up* dan penggunaan kacamata [5]. Ada 2 jenis akuisisi citra dalam pengenalan wajah, yaitu dengan citra 2D dan citra 3D. Citra 3D memberikan informasi geometris lebih rinci dari citra 2D [6]. Sistem pengenalan wajah 3D dapat mengatasi masalah oklusi dan iluminasi dalam sistem pengenalan wajah 2D [7].

Rumusan masalah pada penelitian ini adalah bagaimana merancang sistem pengenalan individu dengan citra wajah 3D. Berdasarkan latar belakang yang dijelaskan, penelitian ini memiliki tujuan untuk menganalisis performansi sistem pengenalan individu dengan citra wajah 3D. Manfaat dari penelitian ini adalah sebagai alternatif sistem pengenalan individu dengan citra wajah 3D.

## 2. Konsep Dasar

Biometrik adalah teknologi yang menggunakan ciri khas dari individu sebagai objek identifikasi untuk mengenali individu. Wajah memiliki ciri khas yang membuat setiap individu berbeda, antara lain: ukuran wajah, ukuran mata, ukuran bibir, jarak antara kedua mata, lebar dan panjang hidung, dan lain-lain. Pengenalan wajah adalah subjek yang begitu berguna dalam implementasi nyata [10].

Pada penelitian [3] menggunakan citra 2D sebagai masukan dalam sistem pengenalan wajah. Namun, citra 2D rentan terhadap kondisi pencahayaan, sudut pengambilan citra, dan ekspresi [3]. Algoritma *Iterative Closest Point* digunakan untuk menentukan korespondensi antar permukaan 3D untuk mengatasi masalah kompensasi dikarenakan wajah yang tidak kaku. Kemudian, wajah dibandingkan menggunakan model statistik *Gaussian Mix-Model* (GMM) dengan citra wajah di *database* RMA 3D yang mendapatkan akurasi pengenalan sebesar 97,33% [8]. Penelitian [9] hanya mendapatkan akurasi sebesar 72,26% menggunakan JST. Hal tersebut dikarenakan bahwa kombinasi metode *supervised learning* seperti JST dan SVM dengan metode analisis regresif, seperti: LDA (*Linear Discriminant Analysis*) kurang begitu cocok. Penelitian [9] juga mengemukakan bahwa perlu adanya kombinasi metode yang lain untuk mendapatkan hasil akurasi pengenalan wajah yang lebih baik.

Dalam penelitian ini akan dirancang sistem pengenalan wajah terhadap database menggunakan *software* dengan memproses citra *points cloud* dari data 3 dimensi yang diambil dengan kamera Kinect yang kemudian dikombinasi dengan metode ICP dan JST sebagai klasifikasi pengenalan wajah individu.

### 2.1. Dasar Pengolahan Citra Digital dan *Depth Image*

Citra digital merupakan suatu fungsi intensitas cahaya  $f(x,y)$ , dengan harga  $x$  dan  $y$  merupakan koordinat spasial dan harga fungsi tersebut. *Depth image* adalah Suatu citra yang menggambarkan kedalaman (*depth*) citra, yaitu jarak permukaan objek dari sudut pandang pengambilan citra. Terdapat 2 jenis *depth image*, yaitu RGB dan *grayscale*.

### 2.2. *Point Cloud*

*Point Cloud* adalah sekumpulan data *point* yang diperoleh dari 3D *scanner*. Biasanya data *point* tersebut berisi informasi mengenai koordinat 3D data tersebut. *Point cloud* merupakan representasi dari sebuah permukaan obyek yang paling sederhana [10].

### 2.3. Kamera Kinect & KScan3D

Kinect merupakan kamera yang dikembangkan oleh Rare di bawah naungan Microsoft. Kinect menggunakan sensor infra merah yang digunakan untuk mendeteksi benda atau gerakan yang ada didepan kamera. Selain itu, Kinect juga alat kamera RGB, sensor kedalaman yang mampu menyajikan citra dalam bentuk 3 dimensi.

KScan3D merupakan rancangan dari LMI Technologies. KScan3D digunakan untuk mengambil citra 3 dimensi. *Software* tersebut dapat digunakan dengan beberapa perangkat seperti: Kinect, Xbox One, and Asus Xtion. Proses *cropping* dan *smoothing* juga dapat dilakukan pada *software* ini. Selain hal tersebut, KScan3D dapat melakukan kombinasi beberapa citra 3D yang diambil.

### 2.4. Iterative Closest Point (ICP)

*Iterative Closest Point* atau yang biasa dikenal dengan ICP merupakan metode ekstraksi yang cara kerjanya melalui pencocokan antar *point clouds* dari sebuah citra. Pencocokan yang biasa dikenal sebagai *Template Matching*. *Template matching* ialah teknik yang digunakan dalam pengolahan citra digital untuk menemukan bagian yang cocok dengan template citra [11]. Kelebihan metode ICP adalah dapat menampilkan informasi yang hilang dari citra 2D. Oleh sebab itu, ICP sering digunakan untuk merekonstruksi citra 2D. Kunci dari algoritma ICP adalah algoritma *corresponding point set registration*. Algoritma tersebut untuk menguraikan *coordinate transformation matrix* dan mencari hubungan antar *point set*.

Metode *quaternion* digunakan untuk menguraikan matriks transformasi di antara dua *point set*. Jika terdapat dua *point set*, *point set* tersebut dapat dibagi menjadi dua bagian, untuk bagian *point target* disebut P, sedangkan untuk bagian *point reference* disebut Q. Adapun syarat-syarat untuk *point set* yang sesuai (*corresponding set point*) [12]:

1. Parameter  $N_P$  merupakan jumlah dari P dan parameter  $N_Q$  merupakan jumlah dari Q, di mana  $N_P = N_Q$
2. Untuk setiap *point* ( $p_i$ ) dari P harus sesuai dengan *point* ( $q_i$ ) dari Q

Metode *quaternion* terdapat vektor rotasi. Vektor rotasi yang seharusnya adalah unit *quaternion*. Vektor rotasi dapat disimbolkan dengan  $q_R$ .

$$q_R = [q_0 \ q_1 \ q_2 \ q_3]^T \quad (1)$$

di mana :  $q_0 > 0$  dan  $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$

Untuk mendapatkan matriks yang paling sesuai dengan dua *point set*. Jarak antara *corresponding point set* harus minimum. Solusi terakhir dari *coordinate transformation* adalah untuk menyelesaikan  $q$ , sehingga dapat dicari nilai minimum dari fungsi objektif.

$$f(q) = \frac{1}{n} \sum_{i=1}^n \|q_i - R(q_R) \cdot p_i - q_T\|^2 \quad (2)$$

Proses dari algoritma ICP adalah :

Langkah 1 : Menentukan target point set (P) dan reference point set (Q)

Langkah 2 : Hitung nilai tengah dari P dan Q

$$\mu_P = \frac{1}{N_P} \sum_{i=1}^{N_P} P_i \quad (3)$$

$$\mu_Q = \frac{1}{N_Q} \sum_{i=1}^{N_Q} Q_i \quad (4)$$

Langkah 3 : Menyusun matriks kovarian dari P dan Q

$$\Sigma_{P,Q} = \frac{1}{N_P} \sum_{i=1}^{N_P} [(p_i - \mu_P)(q_i - \mu_Q)^T] = \frac{1}{N_P} \sum_{i=1}^{N_P} [p_i q_i^T] - \mu_P \mu_Q^T \quad (5)$$

Langkah 4 : menyusun 4x4 matriks simetris dari matriks kovarian

$$H(\Sigma_{P,Q}) = \begin{bmatrix} \text{tr}(\Sigma_{P,Q}) & \Delta^T \\ \Delta & \Sigma_{P,Q} + \sum_{P,Q}^T \text{tr}(\Sigma_{P,Q}) I_3 \end{bmatrix} \quad (6)$$

di dalam matriks diatas,  $I_3$  adalah unit matriks  $3 \times 3$ ,  $\text{tr}(\Sigma_{P,Q})$  adalah *trace* dari  $\Sigma_{P,Q}$

Langkah 5 : tentukan *eigenvalue* dan *eigenvector* dari  $H(\Sigma_{P,Q})$ , cari nilai *eigenvalue* dan *eigenvector*-nya yang terbesar, *eigenvector* adalah vektor rotasi terbaik;

$$q_R = [q_0 \ q_1 \ q_2 \ q_3]^T \quad (7)$$

Langkah 6 : Hitung vektor translasi terbaik;

$$q_T = \mu_Q - R(q_R)\mu_P \quad (8)$$

Dalam Persamaan (8)

$$R(q_R) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 - q_3^2 \end{bmatrix} \quad (9)$$

Langkah 7: diperoleh koordinat vektor transformasi akhir  $q = [q_R | q_T]^T = [q_0 \ q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T$  dan mean square error  $d_{min} = f(q)$ ;

## 2.5. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan sebuah metode yang menggambarkan cara kerja otak manusia dalam merepresentasikan proses pembelajarannya [13]. Dalam JST terdapat beberapa metode pelatihan terbimbing (*Supervised Learning*). Metode tersebut antara lain: *Single Perceptron*, *Multi Perceptron* dan *Backpropagation*.

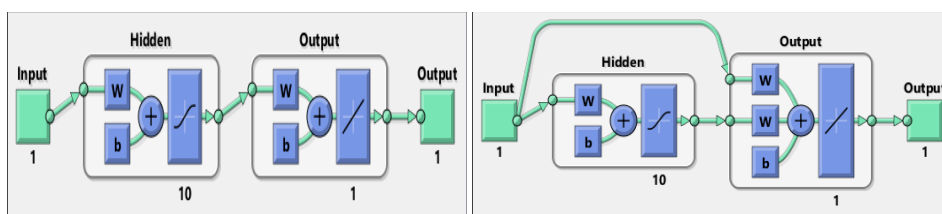
### 1. Arsitektur JST

*Single layer network* memiliki 1 *layer input* dan 1 *layer output* yang terhubung langsung. Pada jaringan ini, nilai *input* langsung diolah menjadi nilai *output* tanpa melalui lapisan tersembunyi. Jaringan ini efisien terhadap waktu dan permasalahan yang sederhana.

*Multi Layer Network* merupakan arsitektur jaringan di mana memiliki beberapa *layer*, antara lain: *input layer*, *hidden layer* dan *output layer*. Jaringan ini sangat baik untuk menyelesaikan permasalahan yang kompleks namun membutuhkan waktu yang lama untuk penyelesaiannya.

### 2. Backpropagation

*Backpropagation* adalah algoritma pembelajaran yang menggunakan *error* sebagai dasar dalam pelatihan dan pemecahan masalah. Jaringan *backpropagation* termasuk dalam jenis *multilayer*. Sehingga, terdapat banyak *layer* terhubung satu sama lain. *Input layer* terhubung dengan *hidden layer*, *output layer* terhubung dengan *hidden layer* [14].



Gambar 1. (a) Arsitektur FFNN (Sumber : <https://www.mathworks.com/help/deeplearning/ref/feedforwardnet.html>); (b) Arsitektur CFNN (Sumber : <https://www.mathworks.com/help/deeplearning/ref/cascadeforwardnet.html>)

a. *Feed forward Neural Network* (FFNN)

*Feed Forward Neural Network* adalah tipe JST yang paling sederhana. FFNN menyediakan kerangka umum untuk mewakili pemetaan fungsi nonlinear antara set variable *input* dan set variable *output*. FFNN terdiri dari beberapa layer dan setiap lapisannya mengambil *input* dari *output* lapisan sebelumnya yang mengalir dalam satu arah. Arsitektur FFNN dapat dilihat pada Gambar 1 (a).

b. *Cascade Forward Neural Network* (CFNN)

*Cascade Forward Neural Network* mirip dengan FFNN. Namun yang membedakan adalah *input* di setiap lapisan meliputi semua *input* dan *output* lapisan sebelumnya [15]. Arsitektur CFNN dapat dilihat pada Gambar 1 (b).

### 3. Perancangan Sistem

Perancangan sistem dalam penelitian ini untuk pemodelan pengenalan individu dari citra wajah 3 dimensi. Pada sistem ini terdapat dua jenis data, yaitu data latih dan data uji. Hasil dari data latih disimpan dalam database yang kemudian ciri dari hasil latih digunakan pada data uji untuk diklasifikasikan dalam pengenalan individu.

Pada bagian ini, akan dibahas alur pembuatan sistem pada setiap tahapannya secara keseluruhan. Perancangan sistem ini menggunakan *software* Matlab untuk pemrosesan citra pengenalan individu berdasarkan citra wajah 3 dimensi. Tahapan perancangan sistem dapat dilihat pada Gambar 2.

1. Akuisisi Citra 3D

Pengambilan citra merupakan tahap awal dalam perancangan sistem ini. Pengambilan citra dilakukan menggunakan kamera Kinect versi 2 dengan bantuan *software* KScan3D. Citra wajah yang diambil adalah tanpa ekspresi diperlakukan sama. Citra yang diambil dari 19 orang terdiri dari 8 pria dan 11 wanita. Setiap orang memiliki 5 citra latih dan 3 citra uji. Satu citra 3D berasal dari 48 foto yang diambil gambar citra 3D dengan kamera Kinect versi 2.

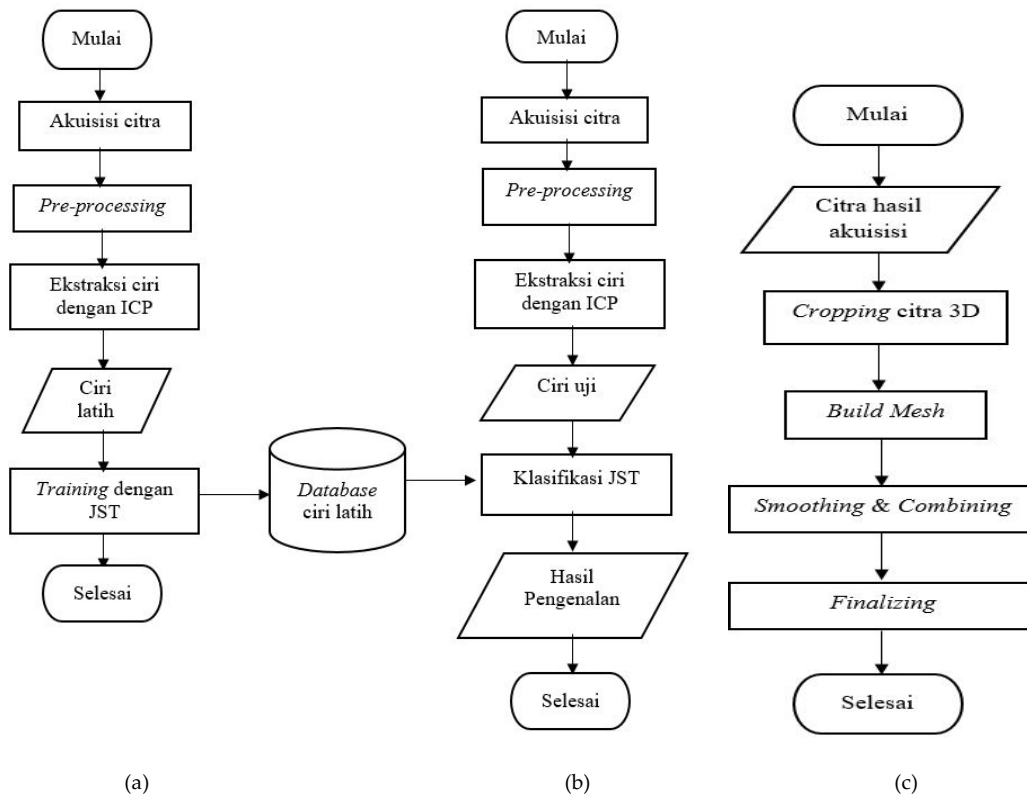
2. *Preprocessing*

Tahap *preprocessing* pada penelitian ini menggunakan bantuan *software* KScan3D. Tahap ini bertujuan untuk memperoleh citra yang baik dengan *noise* dan dimensi yang rendah. Dalam *software* tersebut dilakukan *cropping*, *smoothing*, *combine* dan *finalize*. Keluaran citra dari tahap *preprocessing* berformat polygon file (\*.ply). Diagram alir *Preprocessing* dapat dilihat pada Gambar 2 (c).

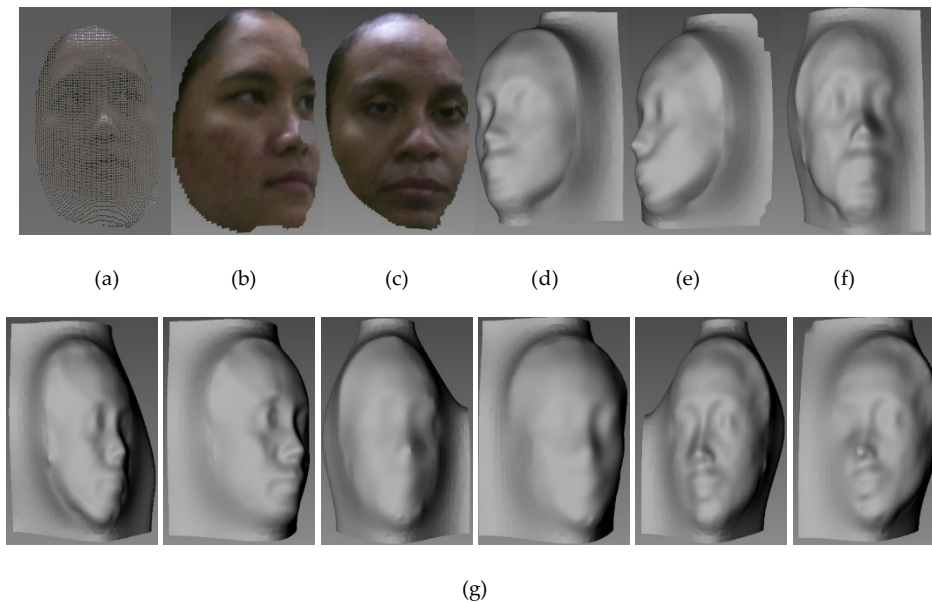
Pengambilan gambar sebanyak 48 foto memiliki hasil yang baik dibandingkan dengan jumlah pengambilan gambar yang lebih sedikit. Pengambilan gambar

dengan jumlah sedikit membuat hasil *output* kurang begitu baik. Namun jika pengambilan gambar dengan jumlah yang membutuhkan waktu *preprocessing* menjadi lebih lama. Jumlah pengambilan gambar sebanyak 48 foto menghasilkan *output preprocessing* yang baik dan waktu yang tidak terlalu lama. Gambar 3 menunjukkan perbandingan *output preprocessing* dari jumlah pengambilan gambar.

*Cropping* citra 3D bertujuan untuk menghilangkan *noise* yang ada pada citra 3D yang dapat memengaruhi hasil ekstraksi ciri. *Cropping* dilakukan secara manual karena *noise* yang terdapat pada setiap citra yang diambil berbeda-beda dan proses *cropping* yang dilakukan untuk mengambil bagian wajah dari setiap individu. Proses *build mesh* bertujuan untuk mengubah *point cloud* menjadi bentuk *mesh geometry*. Bentuk *mesh geometry* akan mempermudah proses *smoothing* dan *combining*. Proses *smoothing & combining* bertujuan untuk menghaluskan tingkat *noise* dan menyatukan beberapa foto bentuk *mesh geometry*. Proses *combining* dilakukan berdasarkan pembagian citra yang telah ditentukan. Parameter yang pembagian citra adalah 3 partisi dan 6 partisi. *Finalizing* bertujuan untuk menyempurnakan bentuk *mesh geometry* yang sudah di *combine*. Citra yang sudah di *finalize*, lalu di *export* ke dalam format polygon file (\*.ply). Gambar 4 (a)-(f) menampilkan hasil *preprocessing* untuk 3 partisi, sedangkan dan Gambar 4 (g) menampilkan hasil *finalize* untuk 6 partisi.



Gambar 2. Diagram Alir (a) Proses latih (b) Proses Uji (c) Preprocessing

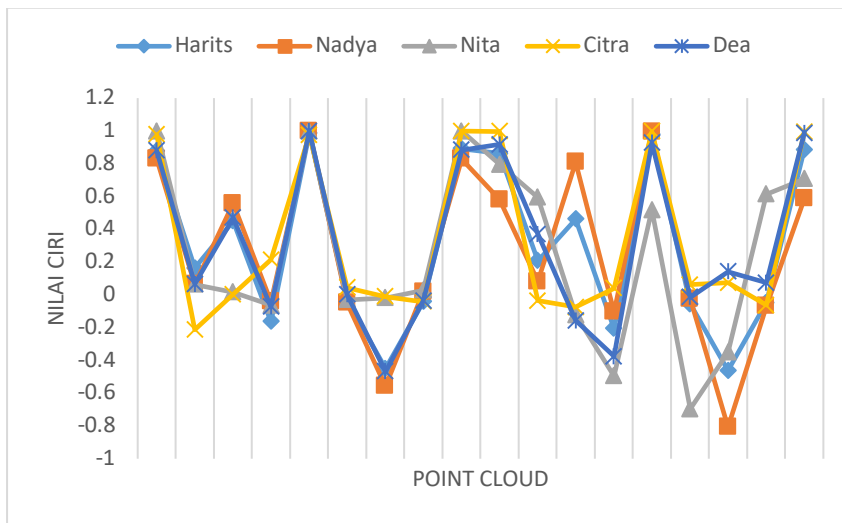


Gambar 4. (a) Hasil *cropping* (b) Hasil *build mesh* (c) Hasil *smoothing & combine* (d) Hasil *finalize* partisi 3 sisi kiri (e) Hasil *finalize* partisi 3 sisi tengah (f) Hasil *finalize* partisi 3 sisi kanan (g) Hasil *finalize* partisi 6

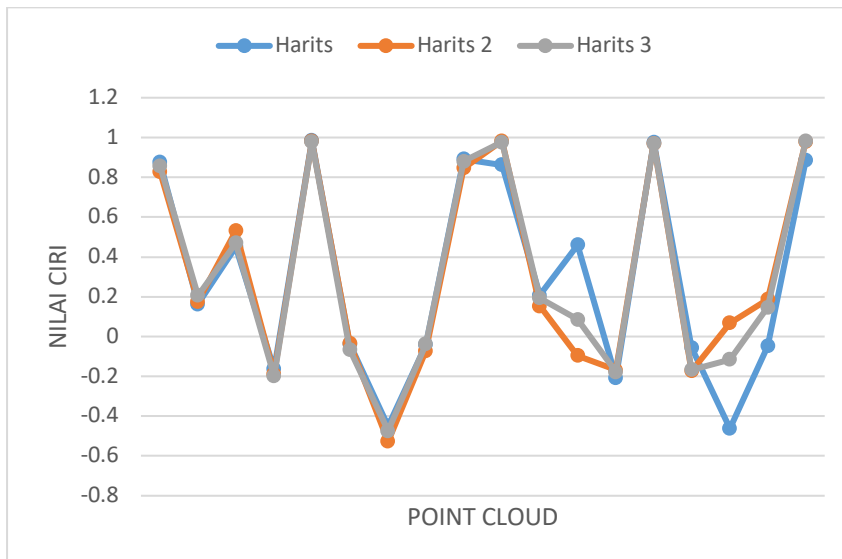
### 3. Ekstraksi Ciri

Ekstraksi ciri pada sistem ini menggunakan metode ICP. Ekstraksi ciri menghasilkan nilai ciri yang beragam dari setiap individu. Keberagaman nilai ciri juga bergantung terhadap jumlah partisi. Semakin banyak jumlah partisi, maka nilai ciri yang diperoleh juga semakin banyak dan beragam. Metode ICP menghasilkan nilai ciri dalam bentuk matriks  $3 \times 3$ , nilai ciri tersebut memiliki perbedaan disetiap datanya. Grafik nilai ciri dari beberapa individu dapat dilihat pada Gambar 5, sedangkan grafik nilai ciri dari satu individu yang sama, dapat dilihat pada Gambar 6.

Berdasarkan Gambar 5, dapat dilihat bahwa setiap individu memiliki nilai ciri yang berbeda. Hal tersebut yang digunakan untuk proses klasifikasi dalam membedakan kelas tiap individu. Semakin besar nilai ciri yang berbeda, maka semakin memudahkan proses klasifikasi dalam membedakan kelas antar individu. Pada Gambar 8, individu yang sama memiliki nilai ciri yang mirip atau hampir sama dengan data masukan yang lain. Hal tersebut akan sangat berpengaruh terhadap akurasi sistem.



Gambar 5. Nilai Ciri Beberapa Individu



Gambar 6. Nilai Ciri dari Individu yang Sama

4. Klasifikasi

Klasifikasi pada perancangan sistem ini menggunakan metode JST. Dalam sistem ini terdapat dua proses klasifikasi, yaitu proses latih dan proses uji. Proses latih dilakukan dengan menggunakan data latih sebagai masukannya dan keluarannya akan disimpan ke dalam *database*. Diagram alir dari proses latih dapat dilihat pada Gambar 2 (a). Sedangkan, proses uji dilakukan dengan cara mencocokkan data uji dan *database*. Diagram proses latih dapat dilihat pada Gambar 2 (b). Hasil klasifikasi adalah sistem mampu mendeteksi individu yang diujikan.

5. *Confusion Matrix*

*Confusion matrix* berisi informasi mengenai hasil prediksi dan data aktual yang dilakukan oleh sistem klasifikasi [16]. Parameter-parameter dalam *confusion matrix* dapat dilihat pada Tabel 1:

Tabel 1. Parameter-Parameter dalam *Confusion Matrix*



		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP ( <i>True Positive</i> )	FP ( <i>False Positive</i> )
	FALSE	FN ( <i>False Negative</i> )	TN ( <i>True Negative</i> )

di mana TP (*True Positive*) merupakan jumlah data yang dideteksi dengan benar; FP (*False Positive*) merupakan jumlah data yang harusnya dideteksi salah, namun oleh sistem dideteksi benar; FN (*False Negative*) merupakan jumlah data yang harusnya dideteksi benar, namun oleh sistem dideteksi salah; TN (*True Negative*) merupakan jumlah data yang dideteksi salah.

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

## 4. Hasil dan Analisis

### 4.1. Pengujian Sistem

Pengujian sistem dilakukan penulis setelah melalui tahapan implementasi rancangan sistem mengenai pengenalan individu berbasis citra wajah 3D. Pengujian sistem bertujuan untuk memperoleh akurasi dari kinerja sistem yang sudah dirancang sebelumnya. Pengujian sistem memiliki beberapa parameter pengujian antara lain: jumlah partisi, banyaknya iterasi, jumlah *hidden layer* dan arsitektur *neural network*.

### 4.2. Hasil Pelatihan Sistem dan Pengujian Sistem

Pada bagian ini dilakukan pelatihan sistem menggunakan data latih yang telah diakuisisi. Data latih yang digunakan dalam pelatihan berjumlah 60 data dari 12 individu. Pelatihan sistem dimaksudkan untuk mencari parameter terbaik yang digunakan dalam pengujian sistem dengan data uji. Sedangkan untuk pengujian sistem dilakukan dengan menguji dari data uji. Terdapat 42 data uji yang berasal dari 18 individu. Di mana 12 individu merupakan data yang dikenali oleh sistem dan 6 individu adalah data yang tidak dikenali oleh sistem

#### 4.2.1. Pengaruh jumlah partisi terhadap performansi sistem

Pelatihan pengaruh perubahan jumlah partisi terhadap beberapa parameter, antara lain: iterasi ke-5, 5 *hidden layer* dan algoritma klasifikasi CFNN. Jumlah partisi yang digunakan adalah 3 partisi dan 6 partisi. Hasil pelatihan pengaruh jumlah partisi terhadap performansi sistem dapat dilihat pada Tabel 2.

Tabel 2 Hasil Pelatihan Pengaruh Jumlah Partisi

Partisi	Akurasi	Rata-rata waktu komputasi (detik)
3	59.70%	20.329
6	77.61%	37.8

Berdasarkan Tabel 2 dapat dilihat bahwa jumlah partisi berpengaruh terhadap akurasi dan waktu komputasi sistem. Akurasi pelatihan tertinggi adalah 77.61% pada 6 partisi. Sehingga, dapat disimpulkan bahwa 6 partisi adalah parameter yang paling cocok digunakan pada sistem ini. Jumlah partisi yang semakin banyak akan menghasilkan nilai

ciri yang semakin banyak dan beragam. Dengan demikian, perbedaan nilai ciri antar individu lebih jelas terlihat.

#### 4.2.2. Pengaruh jumlah iterasi terhadap performansi sistem

Pelatihan pengaruh perubahan jumlah iterasi terhadap beberapa parameter, yaitu: 5 *hidden layer* dan algoritma klasifikasi CFNN yang dikombinasikan dengan parameter terbaik dari skenario pertama, yaitu 6 partisi. Jumlah partisi yang digunakan adalah iterasi ke-5, iterasi ke-25 dan iterasi ke-49. Hasil pelatihan pengaruh jumlah iterasi dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pelatihan Pengaruh Jumlah Iterasi

Iterasi	Akurasi	Rata-rata waktu komputasi (detik)
5	77.61%	37.8
25	65.67%	438.543
49	65.67%	452.78

Berdasarkan Tabel 3 dapat dilihat bahwa jumlah iterasi memiliki pengaruh terhadap akurasi sistem dan waktu komputasi. Hasil terbaik yang didapat dari pengujian skenario 2 adalah pada iterasi ke-5, yaitu sebesar 77.61%. Jumlah iterasi yang sedikit memberikan nilai ciri yang unik dan waktu komputasi yang relatif cepat dibandingkan jumlah iterasi yang relatif banyak. Namun, jika jumlah iterasi sangat sedikit, maka nilai ciri tersebut belum dapat merepresentasikan nilai ciri dari seorang individu.

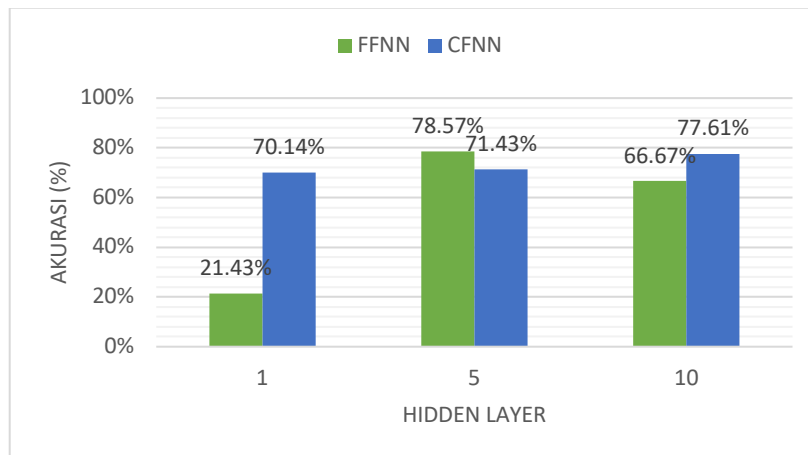
#### 4.2.3. Pengaruh jumlah *hidden layer* dan arsitektur *neural network* terhadap performansi sistem

Pengujian pengaruh perubahan jumlah *hidden layer* terhadap beberapa parameter, antara lain: algoritma klasifikasi CFNN yang dikombinasikan dengan parameter terbaik dari skenario pertama dan kedua, yaitu 6 partisi dan iterasi ke-5. Hasil pelatihan pengaruh jumlah *hidden layer* dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian Pengaruh Jumlah *Hidden Layer* dan Arsitektur *Neural Network*

Hidden layer	Algoritma klasifikasi	Akurasi	Rata-rata waktu komputasi (detik)
1	CFNN	70.14%	37.8310
5		71.43%	37.8504
10		77.61%	37.8694
1	FFNN	21.43%	37.8156
5		78.57%	37.8318
10		66.67%	37.8584

Berdasarkan Tabel 4 dapat dilihat bahwa banyaknya *hidden layer* juga berpengaruh terhadap akurasi, dan berpengaruh terhadap waktu komputasi. Dapat disimpulkan bahwa semakin banyak jumlah *hidden layer* maka semakin tinggi nilai akurasi dan semakin lama waktu komputasi yang dibutuhkan. Selain itu, jenis arsitektur *neural network* yang semakin kompleks dalam penelitian ini adalah CFNN, maka nilai akurasi dalam setiap *hidden layer* menjadi relatif berimbang. Sebaliknya, dalam algoritma FFNN, nilai akurasi terbaik dengan 5 *hidden layer*. Algoritma FFNN lebih cocok digunakan dalam sistem ini dibandingkan dengan CFNN karena algoritma FFNN merupakan algoritma yang paling sederhana dan dapat diimplementasikan di berbagai topik permasalahan. Hasil terbaik yang didapat pada pengujian skenario ini adalah pada 5 *hidden layer* dengan akurasi sebesar 78.57%. Perbandingan akurasi algoritma FFNN dan CFNN berdasarkan jumlah *hidden layer* dapat dilihat pada Gambar 7.



Gambar 7. Perbandingan Akurasi Algoritma FFNN dan CFNN Berdasarkan Jumlah *Hidden Layer*

#### 4.2.4. Confusion Matrix

Selain akurasi dan waktu komputasi, performansi sistem dapat dinilai melalui *confusion matrix*. Nilai-nilai *confusion matrix* didapatkan dari persamaan (10) untuk nilai *recall*, (11) untuk nilai *precision* dan (12) untuk nilai *accuracy*. Hasil pengujian sistem menghasilkan nilai dari parameter-parameter *confusion matrix*.

Tabel 5. Hasil *Confusion Matrix*

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP = 27	FP = 6
	FALSE	FN = 3	TN = 6

Dari persamaan (10), (11) dan (12) diperoleh:

$$Recall = \frac{TP}{TP+FN} = \frac{27}{27+3} = 0.9 = 90\%$$

$$Precision = \frac{TP}{TP+FP} = \frac{27}{27+6} = 0.8181 = 81.81\%$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{27+6}{27+6+6+3} = 0.7857 = 78.57\%$$

## 5. Kesimpulan

Pada penelitian ini, telah dirancang sebuah sistem pengenalan individu dengan citra wajah 3D menggunakan metode ICP untuk ekstraksi ciri dan metode JST untuk klasifikasi. Parameter yang paling cocok untuk sistem ini adalah dengan menggunakan 6 partisi, iterasi ke-5, 5 *hidden layer* dan algoritma FFNN. Dengan parameter tersebut, didapatkan akurasi sebesar 78.57%.

Dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa metode ICP dan JST dapat digunakan untuk sistem pengenalan individu berbasis citra wajah 3D. Namun kurang begitu bagus diimplementasikan dalam sistem pengenalan individu berbasis citra wajah 3D, karena hanya mendapatkan akurasi sebesar 78.57%. Untuk sistem pengenalan individu disarankan memakai metode dengan akurasi diatas 90%.

Penelitian selanjutnya diharapkan dapat memperbaiki akurasi sistem pengenalan individu berbasis citra 3D menggunakan metode yang berbeda, membuat sebuah sistem pengenalan secara *real time*, maupun perbaikan dalam jumlah individu yang menjadi objek pengenalan wajah.

## Daftar Pustaka

- [1] D. E. Kurniawan, K. Adi and A. F. Rohim, "Sistem Identifikasi Biometrika Wajah Menggunakan Metode Gabor KPCA dan Mahalanobis Distance," *Jurnal Sistem Informasi Bisnis*, vol. 01, h. 6-10, 2012.
- [2] A. Budi, Suma'inna and H. Maulana, "Pengenalan Citra Wajah sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA)," *Jurnal Teknik Informatika*, vol. 9, no. 2, h. 166-175, 2016.
- [3] I. G. Barus and R. A. Saragih, "Simulasi Ekstraksi Citra Wajah dengan Deskriptor Global Direct Fractional-Step Linear Discriminant Analysis (DF-LDA) untuk Pengenalan Wajah," *Techné Jurnal Ilmiah Elektroteknika*, vol. 14, no. 1, h. 59-66, 2015.
- [4] J. Choudhary, "Survey of Different Biometrics Techniques," *International Journal of Modern Engineering Research (IJMER)*, vol. 2, no. 5, pp. 3150-3155, 2012.
- [5] M. Hiremath and P. S. Hiremath, "3D Face Recognition Based on Symbolic FDA Using SVM Classifier with Similarity and Dissimilarity Distance Measure," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, no. 4, p. 1756006 (22 pages), 2017.
- [6] B. B. Q. J. W. Sima Soltanpour, "A Survey of Local Feature Methods for 3D Face Recognition," *Elsevier*, vol. 72, pp. 391-406, 2017.
- [7] R. Ajmera, A. Nigam and P. Gupta, "3D Face Recognition using Kinect," *Indian Institute of Technology*, 2014.
- [8] M. N. D. R. G. S. Andrea F. Abate, "2D and 3D face recognition: A survey," *Elsevier*, vol. 28, pp. 1885-1906, 2007.
- [9] S. S. D. B. Vaibhav M. Pathak, "An Efficient Approach for 3D Face Recognition Using ANN Based Classifiers," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 9, pp. 8527-8531, 2015.
- [10] D. S. D. V. Thomas Fabry, "Surface representations for 3D face recognition," *Researchgate.net*, vol. 15, pp. 273-294, 2010.
- [11] R. C. V. Frank B. ter Haar, "A 3D Face Matching Framework," in *IEEE International Conference on Shape Modeling and Applications*, Stony Brook, New York, USA, 2008.
- [12] J. J. G. Z. J. Z. Hao Yang, "An Improved Iterative Closest Points Algorithm," *World Journal of Engineering and Technology*, vol. 3, pp. 302-308, 2015.
- [13] S. Kusmaryanto, "Jaringan Saraf Tiruan Backpropagation untuk Pengenalan Wajah Metode Ekstraksi Fitur Berbasis Histogram," *EECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems)*, vol. 8, no. 2, h. 193-198, 2014.
- [14] S. E. S. A. John Dhanaseely, "Performance Comparison of Cascade and Feed forward Neural Network for Face Recognition system," in *International Conference on Software Engineering and Mobile Application Modelling and Development (ICSEMA 2012)*, Chennai, India, 2012.
- [15] A. Biswas, M.K.Ghose and M. Pandit, "Comparison of Different Neural Network Architectures for Classification of Feature Transformed Data for Face Recognition," *International Journal of Computer Applications (0975 – 8887)*, vol. 96, no. 12, pp. 25-31, 2014.

- [16] Y. T. Handika, "Implementasi Metode Filter Gabor dan Backpropagation Neural network pada Sistem Pengenalan Wajah sebagai Personal Authentication," *Universitas Pendidikan Indonesia*, 2014.
- [17] P. Kamencay, R. Hudec, M. Benco, P. Sykora and R. Radil, "An Efficient P-KCCA Algorithm For 2D-3D Face Recognition Using SVM," *Digital Image Processing and Computer Graphic*, vol. 13, no. 4, pp. 399-406, 2015.

