

SISTEM PENGENALAN WAJAH 3D MENGGUNAKAN ICP DAN SVM

Nadya Viana de Lima¹, Ledy Novamizanti², Eko Susatio³

^{1,2,3}Teknik Telekomunikasi, Universitas Telkom

Email: ¹nadyadelimaa@gmail.com, ²ledyaldn@telkomuniversity.ac.id, ³maharusdi@gmail.com

(Naskah masuk: 03 Januari 2018, diterima untuk diterbitkan: 02 April 2019)

Abstrak

Pengenalan wajah merupakan salah satu teknologi biometrik yang banyak diaplikasikan terutama pada sistem keamanan. Sistem absensi dengan wajah, mengenali pelaku tindak kriminal dengan CCTV adalah beberapa aplikasi dari pengenalan wajah. Efisiensi dan akurasi menjadi faktor utama pengenalan wajah banyak diaplikasikan. Pada penelitian ini, sistem identifikasi diimplementasikan dalam bentuk pengenalan wajah 3 dimensi berbasis *template matching* menggunakan metode *Iterative Closest Point* (ICP) dan klasifikasi *Support Vector Machine* (SVM). *Iterative Closest Point* (ICP) memberikan informasi dimensi dengan meminimalisasi kesalahan antara titik-titik dalam satu tampilan dan titik terdekatnya agar template wajah 3D yang dibuat sesuai dengan citra referensi. Sedangkan SVM adalah metode klasifikasi dengan menentukan kelas citra berdasarkan informasi yang diperoleh dari proses ekstraksi ciri. Hasil akhir dari penelitian ini adalah suatu aplikasi yang mampu melakukan identifikasi pengenalan pola wajah 3D. Berdasarkan *confusion matrix*, diperoleh bahwa sistem ini bekerja dengan *precision* 97,30%, *recall* 100,00%, *accuracy* 97,56% pada pengambilan *frame* citra sebanyak 48, iterasi ke 49, partisi 12, dan menggunakan SVM tipe OAA.

Kata Kunci: *Pengenalan Biometrik, Wajah, 3D, ICP, SVM.*

3D FACE RECOGNITION SYSTEM USING ICP AND SVM

Abstract

Face recognition is a biometric technology that is widely applied especially in the security system. Attendance systems with faces, recognizing criminals with CCTV are some of the applications of face recognition. Efficiency and accuracy are the main factors that face recognition is widely applied. In this study, the identification system was implemented in the form of 3-dimensional face recognition based on template matching using the Iterative Closest Point (ICP) method and Support Vector Machine (SVM) classification. Iterative Closest Point (ICP) provides dimensional information by minimizing errors between points in one view and the closest point so that 3D face templates are made in accordance with the reference image. Whereas SVM is a classification method by determining the image class based on information obtained from the extraction of features. The final result of this study is an application that is able to identify 3D face pattern recognition. Based on the confusion matrix, found that this system works with 97.30% precision, recall 100.00%, 97.56% accuracy in image frame capture as much as 48 iterations to 49, the partition 12, and using the SVM-type OAA.

Keywords: *Face Recognition, Biometric, 3D, ICP, SVM.*

1. PENDAHULUAN

Keamanan merupakan hal yang sangat penting, terutama jika berkaitan dengan data pribadi. Jika keamanan tersebut tidak terjamin, akan mendatangkan kerugian bagi pihak terkait. Berdasarkan data yang dikeluarkan oleh World Internal Security and Police Index (WISP), Indonesia menempati urutan ke-84 negara dengan tingkat keamanan dalam segala bidang (Abdelmottle, 2016). Bidang yang dimaksud juga terkait dengan keamanan dalam negeri dan keamanan identitas setiap individunya. Untuk itu, diperlukan suatu sistem

keamanan yang dapat mengidentifikasi seseorang (Prawestiningtyas, 2009).

Biometrik merupakan karakteristik biologis, baik fisik, ciri-ciri maupun perilaku unik seseorang. Susunan gigi, iris mata, wajah, suara, DNA, bahkan tanda tangan juga termasuk biometrik. Biometrik yang sering digunakan dalam identifikasi dan verifikasi seseorang adalah pengenalan wajah (face recognition). Beberapa aplikasi dalam dunia nyata contohnya sistem absensi dengan wajah, *lock screen* telepon genggam menggunakan wajah, dan mengenali pelaku tindak kriminal dengan CCTV. Faktor utama pengenalan wajah menjadi banyak

diaplikasikan khususnya dalam bidang sistem keamanan karena tingkat efisiensi dan akurasi yang baik. Banyak informasi yang bisa didapatkan dari sebuah wajah, baik secara statis maupun dinamis, misalnya warna kulit, struktur tulang wajah, jenis kelamin dan ekspresi wajah. Berdasarkan hal tersebut, wajah menjadi obyek yang banyak merepresentasikan kondisi dan mempunyai karakteristik yang dapat dijadikan sebagai bahan penelitian untuk citra digital.

Pada penelitian sebelumnya, biometrik wajah diterima dengan baik di industri ini (Zhang & Lu, 2012). Namun, masih ada beberapa tantangan yang harus dipertimbangkan saat seseorang merancang sistem pengenalan wajah. Pendekatan klasik yang memanfaatkan foto 2D harus menghadapi iluminasi dan variasi pose. Banyak metode yang sudah diajukan dalam beberapa tahun terakhir. Namun masih belum dapat mencapai akurasi yang lebih baik (Ajmera *et al.*, 2014). Sehingga, untuk mencapai akurasi yang lebih baik, perlu diperluas algoritma pada wajah 3D. Obyek 3D yang dimaksud adalah setiap titik yang membentuk obyek dimana ketika diproses memiliki 3 sumbu koordinat. Namun, salah satu keterbatasan utama metode tersebut adalah tingginya biaya sensor wajah 3D (Mracek *et al.*, 2007).

Salah satu alternatif yang menjanjikan untuk metode algoritma 3D adalah dengan menggunakan perangkat Kinect dimana harganya cukup murah, namun tetap mampu menangkap secara tepat informasi yang mendalam. Pemilihan kamera kinect juga didasarkan atas keunggulan teknologi yang dimiliki dibandingkan dengan kamera yang beredar dipasaran, yaitu tersedia fitur *skeletal tracking* yang berfungsi melacak titik sendi utama tubuh manusia. *Skeletal tracking* tidak lepas dari penggunaan sensor kedalaman (depth sensor). Kinect memetakan objek *frame* gambar manusia yang diambil dari posisi berbeda-beda guna mendapatkan data sebuah area wajah 3D (Basir dkk, 2017).

Metode *Iterative Closest Point* (ICP) pada penelitian sebelumnya digunakan untuk membuat korespondensi antara permukaan 3D supaya mengkompensasi masalah karena sifat wajah yang tidak kaku. Kemudian, setelah proses registrasi selesai, informasi pada wajah di ekstrak menggunakan model statistik *Gaussian Mix-Model* (GMM). Berdasarkan database RMA 3D, diperoleh akurasi ICP dengan tingkat pengenalan 97,33% (Abate *et al.*, 2007). Klasifikasi SVM pada penelitian pengenalan wajah, digunakan untuk komputasi fitur berbasis FDA yang bersifat simbolik dan memperhitungkan citra wajah 3D dan variasi wajah, dimana hasil eksperimen menghasilkan kinerja 99.90% menggunakan SVM (Hiremath, 2017).

Meskipun sudah terdapat program berbasis *image processing* untuk identifikasi personal berdasarkan pola pengenalan wajah 3D, namun belum ada satupun yang mengimplementasikan *Iterative Closest Point* dan *Support Vector Machine*.

Kedua metode ini memiliki keunggulan-keunggulan yang membuatnya dapat digunakan dimana ICP dapat membentuk wajah dari variasi pose. Proses iterasi dilakukan sampai citra tersebut dapat menyamai citra referensi. Kemudian SVM hanya menyimpan sebagian kecil dari data latih dalam setiap iterasi pelatihannya. Meskipun waktu pelatihan SVM umumnya lambat, tetapi metode ini sangat akurat karena kemampuannya untuk menangani model-model *nonlinear* yang kompleks. Pada penelitian ini, dirancang sebuah sistem pengenalan wajah 3D berbasis kamera *kinect* menggunakan metode *Iterative Closest Point* (ICP) dan klasifikasi *Support Vector Machine* (SVM).

2. METODE PENELITIAN

2.1 Template Matching dengan ICP

Iterative Closest Point (ICP) menyatakan dua *point cloud* A dan B, yang terdiri dari masing-masing M dan N point dalam R^3 (Afriliana, 2015). Tujuan utama dari ICP adalah ketika informasi tidak lengkap atau ada informasi yang hilang dari suatu obyek, maka sistem akan melakukan penambahan dimensi informasi dengan meminimalisasi *Mean Square Error* (MSE) antara poin titik-titik dalam satu tampilan dan titik terdekatnya. Registrasi *point set* yang sesuai (Corresponding point set registration), adalah hal yang penting dalam algoritma ICP. Algoritma juga menguraikan koordinat matrik transformasi dan mencari hubungan atau keterkaitan antar *point set*. Adapun pada ICP, metode *quaternion* digunakan untuk menguraikan matriks transformasi diantara 2 *point set*. *Point set* dapat dibagi menjadi dua, yaitu *point set* P sebagai data referensi dan Q sebagai data uji. Beberapa syarat dalam penentuan *point set* yang sesuai, antara lain (Mujib dkk, 2018):

1. Jumlah P (N_p) harus sama dengan jumlah Q (N_q)
2. Untuk setiap poin (p_i) dari P harus sesuai dengan point (q_i) dari Q.

Vektor rotasi merupakan unit *quaternion* dan disimbolkan dengan q_r

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

dimana, $q_0 \geq 0$ dan

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (2)$$

Jarak antar *corresponding point set* haruslah minimum untuk mendapatkan matriks yang sesuai dengan 2 *point set*. Kemudian dapat dicari nilai minimum dari sebuah fungsi objektif untuk menyelesaikan nilai q.

$$f(q) = \frac{1}{n} \sum_{i=1}^n \|q_i - R(q_r) \cdot p_i - q_T\|^2 \quad (3)$$

Langkah- langkah pada ICP antara lain:

Pertama : Menentukan target *point set* (P) dan *reference point set* (Q)

Kedua : Hitung nilai tengah dari P dan Q

$$\mu_P = \frac{1}{N_P} \sum_{i=1}^{N_P} P_i \quad (4)$$

$$\mu_Q = \frac{1}{N_Q} \sum_{i=1}^{N_Q} Q_i \quad (5)$$

Ketiga : Menyusun matriks *covariance* dari P dan Q

$$\begin{aligned} \Sigma_{P,Q} &= \frac{1}{N_P} \sum_{i=1}^{N_P} [(p_i - \mu_P)(q_i - \mu_Q)^T] \\ &= \frac{1}{N_P} \sum_{i=1}^{N_P} [p_i q_i^T] - \mu_P \mu_Q^T \end{aligned} \quad (6)$$

Keempat : Menyusun 4x4 matriks simetris dari matriks *covariance*

$$H(\Sigma_{P,Q}) = \begin{bmatrix} \text{tr}(\Sigma_{P,Q}) & \Delta^T \\ \Delta & \Sigma_{P,Q} + \Sigma_{P,Q}^T \text{tr}(\Sigma_{P,Q}) I_3 \end{bmatrix} \quad (7)$$

Pada formulasi matriks tersebut, I_3 merupakan unit matriks 3x3, dan $\text{tr}(\Sigma_{P,Q})$ adalah *trace* dari $\Sigma_{P,Q}$

Kelima : Penentuan nilai *eigen value* dan *eigen vector* dari $H(\Sigma_{P,Q})$, dimana melihat nilai *eigen value* terbesar dan *eigen vector* dengan rotasi terbaik.

$$q_R = [q_0 \ q_1 \ q_2 \ q_3]^T \quad (8)$$

Keenam : Hitung vektor translasi terbaik

$$q_T = \mu_Q - R(q_R)\mu_P \quad (9)$$

Sesuai dengan persamaan 9, maka didapatkan matriks seperti berikut,

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (10)$$

Ketujuh : Diperoleh koordinat vektor transformasi akhir dengan persamaan sebagai berikut.

$$q = [q_R | q_T] = [q_0 \ q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T \quad (11)$$

dan *mean square error* (MSE)

$$d_{\min} = f(q) \quad (12)$$

2.2 Support Vector Machine (SVM)

Metode klasifikasi *Support Vector Machine* (SVM) merupakan algoritma yang bekerja menggunakan pemetaan *nonlinear* untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi (Thome, 2012). Dimensi baru akan mencari *hyperplane* untuk memisahkan secara *linear* dan pemetaan *nonlinear* yang tepat ke dimensi yang cukup tinggi. Sehingga, kedua kelas tersebut dapat dipisahkan dengan *hyperplane*. *Hyperplane* tersebut diperoleh dengan menggunakan *support vector* dan *margin*.

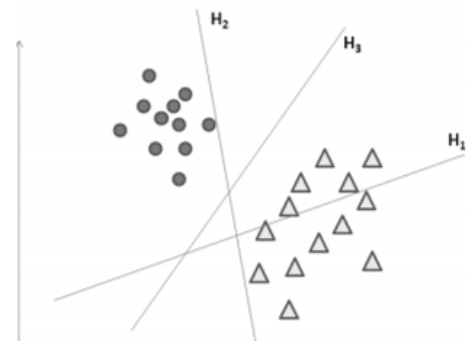
Konsep dari SVM yaitu bekerja dengan baik pada kumpulan data dengan dimensi tinggi. SVM juga menggunakan teknik *kernel* yang memetakan data asli dari dimensi asalnya menjadi dimensi lain yang relatif lebih tinggi (Thome, 2012). SVM hanya menyimpan sebagian kecil dari data latih yang digunakan pada saat klasifikasi. Hal tersebut yang menjadi kelebihan SVM karena tidak semua data latih dilibatkan dalam setiap iterasi pelatihannya.

Meskipun waktu pelatihan SVM kebanyakan lambat tetapi metode ini sangat akurat karena kemampuannya untuk menangani model-model *nonlinear* yang kompleks. Penerapan SVM dapat juga digunakan sebagai *pattern recognition* antara lain deteksi tulisan tangan, pengenalan objek, identifikasi suara, dan lain-lain.

Secara umum, karakteristik dari SVM, yaitu (Nurajizah, 2013):

1. *Support Vector Machine* merupakan *linear classifier*.
2. Untuk mendapatkan pola yang bisa dikenali, maka dilakukan transformasi data pada input *space* ke ruang yang berdimensi lebih tinggi. Optimisasi dapat dilakukan pada ruang vektor tersebut.
3. SVM menerapkan strategi *Structural Risk Minimization* (SRM).
4. SVM pada dasarnya hanya mampu menangani klasifikasi dua kelas.

Support Vector Machine adalah algoritma komputasi yang membangun *hyperplane* atau *set hyperplanes* dalam ruang dimensi tinggi atau tak terbatas. SVM banyak digunakan untuk klasifikasi, regresi, dan lainnya. Pemisahan antara dua kelas yang terpisah secara *linear* dapat dicapai oleh *hyperplane* apapun yang tidak memberikan kesalahan klasifikasi pada semua titik data dari kelas yang dipertimbangkan. Misalnya semua poin milik kelas A diberi label +1, dan semua poin milik kelas B diberi label -1. Pendekatan ini disebut klasifikasi *linier*. Namun ada banyak *hyperplanes* yang mungkin mengklasifikasikan *set* data yang sama, seperti yang dapat dilihat pada Gambar 1. SVM adalah pendekatan yang bertujuan untuk menemukan pemisah *hyperplane* terbaik, yaitu *hyperplane* yang menyediakan jarak *margin* tertinggi antara titik terdekat dari dua kelas, disebut fungsional batas. Pendekatan ini, secara umum menjamin bahwa semakin besar *margin* maka semakin rendah kesalahan generalisasi dari *classifier*.



Gambar 1. Pemisahan *Hyperplanes*

Pada Gambar 1, dapat dilihat bahwa H1 tidak memisahkan dua kelas; H2 terpisah tetapi dengan *margin* yang sangat kecil antar kelas dan H3 memisahkan kedua kelas dengan *margin* yang jauh lebih baik daripada H2. *Margin* merupakan jarak antara *hyperplane* tersebut dengan data terdekat dari

masing-masing kelas. Jika terdapat *hyperplane* seperti demikian, maka jelas bahwa *margin* memberikan batas pemisahan terbaik antar dua kelas, *hyperplane* tersebut dikenal sebagai *hyperplane* margin maksimum dan *classifier* linier yang demikian dikenal sebagai *classifier margin* maksimum.

Multiclass SVM bertujuan untuk menetapkan label ke beberapa elemen himpunan terbatas berdasarkan pada set *SVM linear* atau *non-linear* dasar. Caranya yaitu dengan mengurangi masalah multi-kelas menjadi beberapa masalah biner (13 – 16).

$$\frac{\partial lp}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i \quad (13)$$

$$\frac{\partial lp}{\partial w} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \quad (14)$$

$$\alpha_i [y_i (w \cdot x_i + b) - 1] \quad (15)$$

$$\alpha_i \geq 0, i = 1, 2, \dots, N \quad (16)$$

Kemudian sebagai klasifikasi biner, diasumsikan untuk menghasilkan fungsi keluaran yang memberikan nilai relatif besar untuk contoh-contoh yang termasuk kelas positif dan nilai-nilai yang relatif kecil untuk contoh-contoh yang dimiliki oleh kelas negatif. Dua metode umum untuk membangun pengklasifikasi biner tersebut adalah dimana setiap pengklasifikasi dilatih untuk dapat membedakan (Thome, 2012):

1. Salah satu label terhadap semua label lainnya, dikenal sebagai one-against-all (OAA)
2. Setiap pasangan kelas, dikenal sebagai one-against-one (OAO).

Klasifikasi untuk kasus OAA, yaitu pemenang-mengambil semua strategi, dimana *classifier* dengan fungsi *output* tertinggi adalah kelas yang dimaksud. Untuk kasus klasifikasi OAO dilakukan dengan strategi pemungutan suara max-win, dimana setiap *classifier* memberikan perumpamaan ke salah satu dari dua kelas. Suara untuk kelas yang ditugaskan meningkat ke satu suara, dan akhirnya kelas dengan lebih banyak suara adalah kelas yang dimaksud.

(i) Metode SVM - OAA

OAA multiclassifier adalah komponen dengan sejumlah pengklasifikasi biner, satu untuk setiap kelas. Menggunakan strategi *Winner-Takes-All*, setiap *classifier* biner dilatih mengambil contoh-contoh dari salah satu kelas sebagai positif dan contoh-contoh dari semua kelas lainnya sebagai negatif. Hasil *multiclassifier* diaktifkan untuk kelas yang *classifier* binernya memberikan *output* terbesar di antara semuanya. Secara formal, diberi vektor a dengan *output* dari pengklasifikasi biner, *multiclassifier* menghasilkan vektor $Y_i = (y_1, \dots, y_k)$, dengan cara berikut (17):

$$i^* = \arg \max_{i=1 \dots k} \{a_i\} \quad (17)$$

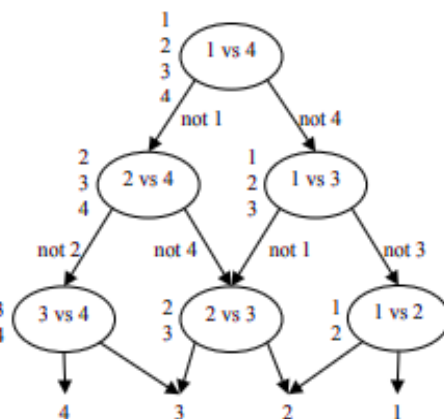
$$Y_i = \begin{cases} +1 & \text{jika } i = i^*, \\ -1 & \text{lainnya} \end{cases} \quad i = 1 \dots k$$

dimana k mempresentasikan jumlah kelas yang ada.

Pada Gambar 2 diilustrasikan bagaimana cara kerja dari *SVM OAA* dalam menemukan kelas terbaik dari 4 kelas. Keadaan setiap daftar kelas setara untuk setiap *node* yang telah ditampilkan di sebelah simpul.

Tabel 1. Contoh SVM biner OAA.

$y_i = +1$	$y_i = -1$	Fungsi keputusan
Kelas 1	Bukan kelas 1	$f^1(x_d) = w^1 \cdot x_d + b^1$
Kelas 2	Bukan kelas 2	$f^2(x_d) = w^2 \cdot x_d + b^2$
Kelas 3	Bukan kelas 3	$f^3(x_d) = w^3 \cdot x_d + b^3$
Kelas 4	Bukan kelas 4	$f^4(x_d) = w^4 \cdot x_d + b^4$



Gambar 2. Cara kerja SVM OAA

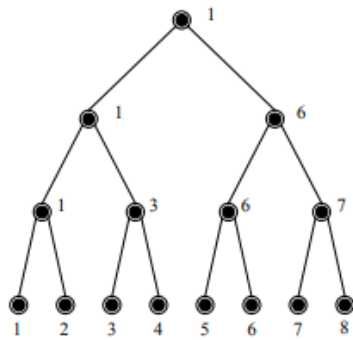
(ii) Metode SVM - OAO

OAO *multiclassifier* membangun satu *classifier* biner untuk setiap pasangan kelas yang berbeda. Untuk X kelas, sejumlah $X * (X-1) / 2$ pengklasifikasi biner dibangun. Klasifikasi biner A_{ij} dilatih mengambil contoh-contoh dari kelas i sebagai positif dan contoh-contoh dari kelas j sebagai negatif. Untuk contoh M yang baru, jika *classifier* A_{ij} diklasifikasikan sebagai kelas i , maka dikenal sebagai kelas i dan ditambahkan satu kelas. Jika tidak, suara untuk kelas j bertambah satu. Setelah masing-masing $X*(M-1)/2$ pengklasifikasi biner memberikan suaranya, strategi memberikan contoh saat ini ke kelas dengan jumlah suara terbanyak.

Tabel 2. Contoh biner SVM OAO

$y_i = +1$	$y_i = -1$	Fungsi keputusan
Kelas 1	Kelas 2	$f^{12}(x_d) = w^{12} \cdot x_d + b^{12}$
Kelas 1	Kelas 3	$f^{13}(x_d) = w^{13} \cdot x_d + b^{13}$
Kelas 1	Kelas 4	$f^{14}(x_d) = w^{14} \cdot x_d + b^{14}$
Kelas 2	Kelas 3	$f^{23}(x_d) = w^{23} \cdot x_d + b^{23}$
Kelas 2	Kelas 4	$f^{24}(x_d) = w^{24} \cdot x_d + b^{24}$
Kelas 3	Kelas 4	$f^{34}(x_d) = w^{34} \cdot x_d + b^{34}$

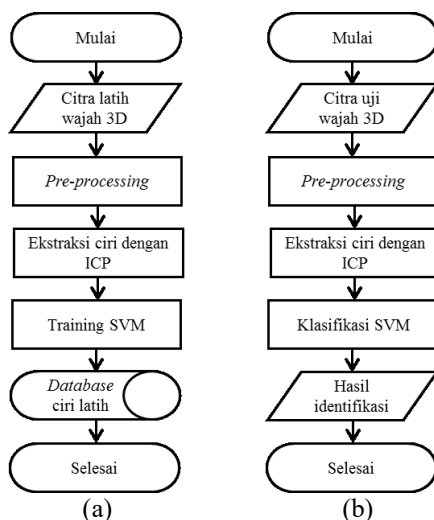
Pada Gambar 3, terdapat 8 kelas untuk pohon biner tersebut, untuk setiap data uji dibandingkan dengan masing-masing dua pasangan, dan pemenang akan diuji ditingkat atas sampai puncak pohon tercapai.



Gambar 3 Cara kerja SVM one-against-one

2.3 Diagram Blok Sistem

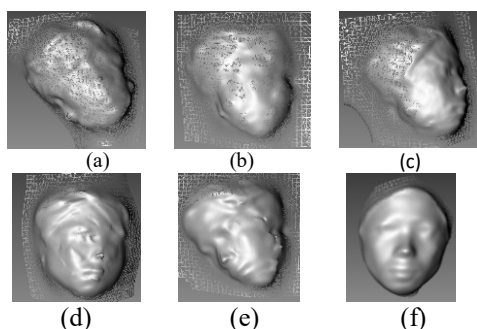
Secara keseluruhan, gambaran umum dari rancangan blok sistem pada penelitian ini adalah sebagai berikut:



Gambar 4. Diagram Alir Perancangan Sistem, (a) Proses Latih, (b) Proses uji

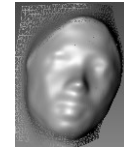
2.4 Akuisisi Citra

Proses akuisisi atau pengambilan citra dilakukan dengan menggunakan kamera Kinect versi 2. Jumlah individu sebanyak 20 orang, yang terdiri dari 8 laki-laki dan 12 perempuan. Kemudian, tiap individu diambil 8 sampel citra, yaitu 4 citra latih dan 4 citra uji. Pengambilan citra dilakukan dengan mengambil gambar bagian wajah dari sisi kiri ke kanan sampai diperoleh pola wajah yang jelas.



Gambar 5. Pengambilan Wajah (a) 3 kali, (b) 6 kali, (c) 8 kali, (d) 24 kali, (e) 36 kali, (f) 60 kali

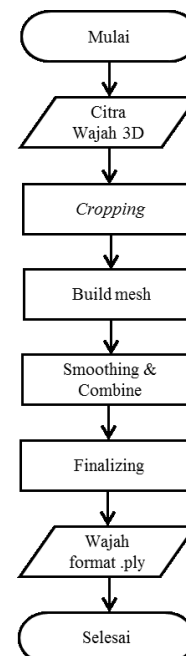
Semakin banyak jumlah *frame* yang diambil, maka semakin baik pula citra gambarnya. Untuk hasil *frame* ke 36 hingga 60 diperoleh hasil yang sangat baik. Oleh karena itu, penulis mengambil citra dengan *frame* sebanyak 48. Hasil penggabungan untuk ke 48 *frame* dapat dilihat pada Gambar 6.



Gambar 6. Pengambilan wajah 48 kali.

2.5 Pemrosesan Awal (*Preprocessing*)

Preprocessing adalah proses awal yang dilakukan sebelum citra diproses lebih lanjut. Tujuan dari dilakukannya proses ini adalah penyeragaman ukuran citra, dan meningkatkan kualitas citra baik untuk citra latih maupun uji. Pada tahapan *3D Face Cropping*, bertujuan untuk memotong bagian wajah dengan menggunakan bantuan *software* KScan3D. Tahapan ini dilakukan manual dengan membuang bagian - bagian selain wajah. Hasil *capture* sebanyak 48 buah akan dipartisi menjadi 3,6,7,12 bagian, sehingga pada tahap akhir terbentuk beberapa data dalam format *file* polygon (*.ply). Tahapan *preprocessing* dapat dilihat pada Gambar 7.

Gambar 7. Diagram alir *preprocessing*

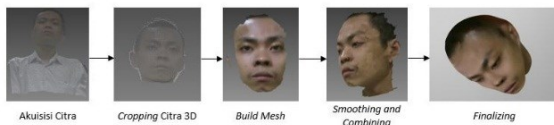
Berikut adalah langkah-langkah *preprocessing*:

1. *Cropping*: Hasil citra 3D mencakup bagian kepala hingga badan bagian atas. Oleh karena itu, dilakukan *cropping* secara manual untuk mendapatkan citra yang hanya terdiri dari bagian lingkaran wajahnya saja.
2. *Build Mesh*. Pada tahap ini, citra wajah yang terdiri dari 48 sampel wajah dipartisi sesuai

dengan jumlah yang diinginkan. *Build Mesh* bertujuan untuk mengkonversi *point cloud* menjadi *mesh geometry* sehingga citra 3D bisa dapat diproses ditahap selanjutnya.

3. *Smoothing and Combining*. Semua citra yang telah terbentuk, menuju ke tahap berikutnya yakni *smoothing and combining*. Tahap ini bertujuan untuk memperbaiki bentuk wajah dimana pada saat tahap sebelumnya, masih terdapat bagian wajah yang hilang.
4. *Finalizing*. Hasil akhir dari proses, disimpan ke dalam format *.ply.

Hasil *preprocessing* dapat dilihat pada Gambar 8.



Gambar 8. Hasil implementasi *Preprocessing*

2.6 Ekstraksi Ciri

Ekstraksi ciri merupakan tahapan untuk mendapatkan informasi yang mendasari ciri dari wajah 3D. Dari hasil *preprocessing*, wajah dipartisi menjadi beberapa bagian yakni 3, 6, 7, 12. Adapun metode yang digunakan pada ekstraksi ciri ini adalah *Iterative Closest Point* (ICP), yaitu prosedur iteratif meminimalisasi *mean square error* (MSE) antara poin dan titik terdekat.

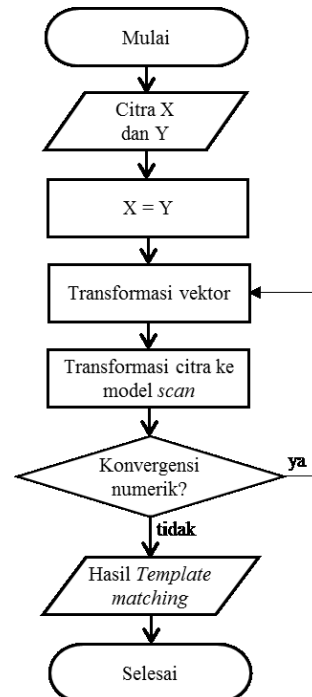
Berikut langkah-langkah untuk menghasilkan *template match*:

1. Masukkan X sebagai data 1 (data referensi) dan Y sebagai data 2 (data target). Untuk data target, disesuaikan dengan jumlah partisi yang diinginkan. Tujuannya untuk menemukan transformasi *vector* yang dapat meminimalkan jarak antara kedua set titik tersebut. Pada tiap partisi, tentukan jumlah iterasi yang diinginkan. Iterasi dilakukan dengan menambahkan beberapa titik *point* pada bagian yang kosong sehingga menjadikan gambar lebih baik.
2. Tentukan vektor transformasi.
3. Tentukan titik baru pada iterasi selanjutnya. Prosedur ini bergantian dan diiterasi sampai konvergensi (yaitu stabilitas kesalahan minimal).
4. Diperoleh vektor transformasi akhir.

Output dari ekstraksi ciri ICP disimpan ke dalam sebuah *template match*, untuk dijadikan sebagai ciri pengenalan wajah. Diagram alir dari tahapan ekstraksi ciri dapat dilihat pada Gambar 9.

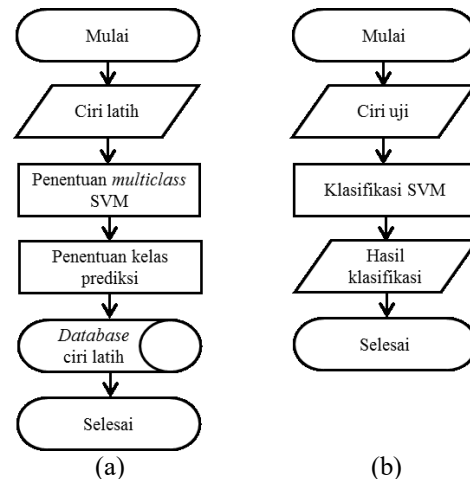
2.7 Klasifikasi Support Vector Machine

Tujuan dari proses klasifikasi menggunakan SVM adalah untuk menentukan kelas pengenalan wajah berdasarkan yang didapat dari proses ekstraksi ciri.



Gambar 9. Diagram Alir Tahapan Ekstraksi Ciri

Pada proses klasifikasi, dibagi menjadi klasifikasi SVM untuk data latih dan klasifikasi SVM untuk data uji. Setelah melewati proses ekstraksi ciri, selanjutnya dilakukan proses klasifikasi SVM untuk data latih dan data uji seperti terlihat pada Gambar 10.



Gambar 10. Blok Diagram Klasifikasi (a) Proses Latih, (b) Proses Uji

Klasifikasi SVM untuk data latih dimulai dengan masukan citra latih hasil ekstraksi ciri ICP, kemudian menentukan metode *multiclass* SVM yang digunakan yaitu *one-against-one* (OAO) atau *one-against-all* (OAA). Selanjutnya, dilakukan proses pelatihan data yang diambil dari *folder* yang sudah didefinisikan sebagai data latih. Terakhir dilakukan proses pengujian data yang membandingkan dengan data citra latih.

2.8 Waktu Performansi

Waktu komputasi merupakan waktu yang diperlukan sistem dalam mengolah data masukan hingga didapatkan *output* yang diinginkan.

$$\text{waktu komputasi} = \text{waktu selesai} - \text{waktu mulai} \quad (18)$$

2.9 Confusion Matrix

Confusion Matrix adalah matriks yang berisi informasi tentang hasil prediksi klasifikasi dan data aktual yang dilakukan oleh sistem klasifikasi. *Confusion matrix* dapat dilihat pada Tabel 3 berikut.

Tabel 3 Confusion Matrix

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive) Correct Result	FP (False Positive) Unexpected result
	FALSE	FN (False Negative) Missing Result	TN (True Negative) Correct Absence of Result

Confusion Matrix dapat dihitung dengan mencari nilai *precision*, *recall* dan *accuracy*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* adalah tingkat keberhasilan sistem dalam menentukan kembali sebuah informasi dan akurasi didefinisikan untuk tingkat kedekatan antara nilai prediksi dengan nilai aktual.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (19)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (20)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN+FP+FN} \quad (21)$$

3. Hasil Pengujian Sistem

Pengujian sistem dilakukan dengan mengubah parameter ekstraksi ciri berupa jumlah *point cloud* dan klasifikasi SVM berupa OAA dan OAO. Pada pengujian ini, akan dianalisis seberapa besar pengaruh perubahan parameter terhadap performansi sistem. Skenario pengujian sistem adalah sebagai berikut:

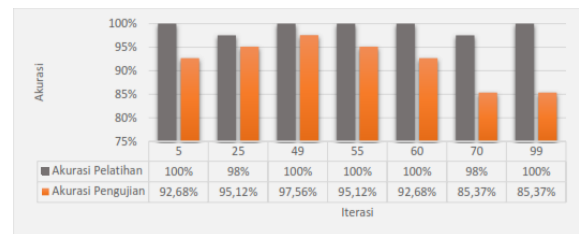
1. Untuk mengetahui banyaknya iterasi yang paling tepat digunakan pada proses ekstraksi ciri. Dilakukan pengujian terhadap beragam nilai iterasi yakni 5, 25, 49, 55, 60, 70 dan 99 untuk melihat nilai iterasi yang paling banyak mendekati konvergen pada tiap citra. Pada tahap ini digunakan klasifikasi OAA dan jumlah partisi 3.
2. Untuk mengetahui banyaknya partisi yang paling tepat digunakan pada proses ekstraksi ciri. . Dilakukan pengujian terhadap beragam perubahan nilai *point cloud* (partisi) yaitu 3, 6, 7

dan 12. Pada pengujian ini, digunakan jumlah iterasi yang mendekati konvergen terbanyak pada skenario 1 dan menggunakan klasifikasi SVM OAA.

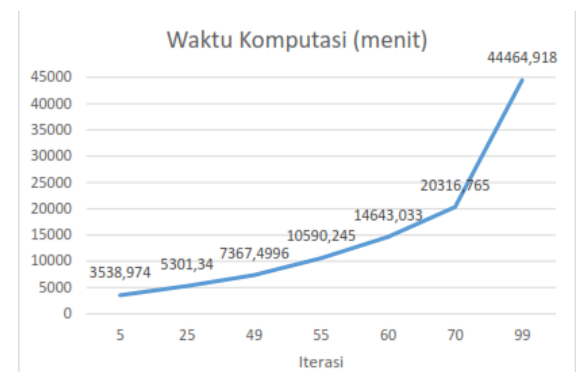
3. Untuk mengetahui penggunaan parameter SVM multiplikasi yang paling tepat digunakan. Dilakukan pengujian terhadap SVM *one-against-one* dan *one-against-all*. Pada pengujian ini digunakan jumlah iterasi terbaik dari skenario 1 dan jumlah partisi terbaik dari skenario 2.

3.1 Pengaruh Jumlah Iterasi pada kekonvergenan Citra

Pada bagian ini, akan dilakukan pengujian sistem menggunakan skenario 1, dimana akan dicari nilai iterasi terbaik yang menunjukkan tingkat konvergen dan koresponden yang terbaik saat mencari ciri citra.



Gambar 11. Hasil Pengujian Skenario 1



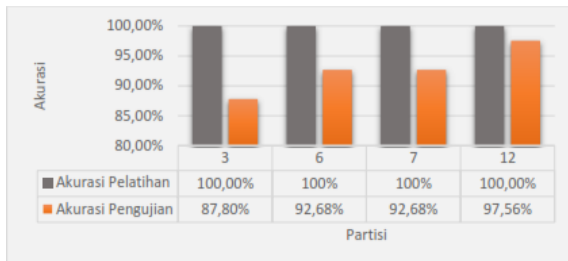
Gambar 12. Waktu Komputasi Skenario 1

Berdasarkan Gambar 11 dan 12, diperoleh bahwa jumlah iterasi pada proses ekstraksi ciri dapat mempengaruhi akurasi pengujian. Akurasi pengujian tertinggi diperoleh saat jumlah iterasi ke-49 yakni 97,56%. Sehingga, jumlah iterasi yang paling tepat digunakan yaitu pada iterasi ke-49. Setiap metode klasifikasi, khususnya SVM pada dasarnya membutuhkan ciri yang khas antar citra agar tidak terjadi kesalahan dalam melakukan pengenalan. Semakin banyak dilakukan iterasi maka citra yang dihasilkan semakin unik, dan khas. Namun tidak disarankan memilih jumlah iterasi terlalu banyak karena memungkinkan terjadinya *oversmooth*, yaitu suatu kondisi dimana akan menghasilkan *output* yang terlalu halus permukaan wajahnya, sehingga membuat wajah tampak sama. Ketika melakukan iterasi maka terdapat penambahan poin atau titik

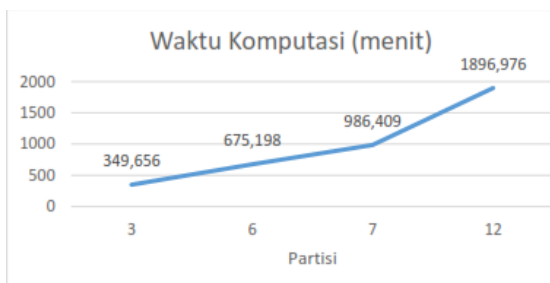
untuk mengisi kekosongan pada permukaan wajah tersebut.

3.2 Pengaruh Jumlah Partisi terhadap Performansi Sistem

Pada bagian ini, dilakukan pengujian sistem menggunakan skenario 2.



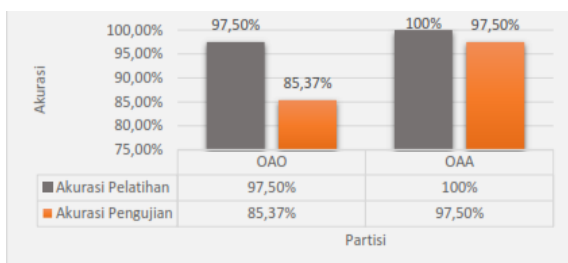
Gambar 13. Hasil Pengujian Skenario 2



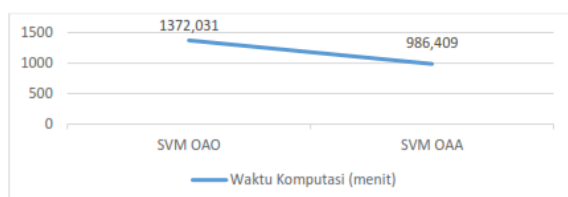
Gambar 14. Waktu Komputasi Skenario 2

Berdasarkan Gambar 13 dan 14, diperoleh bahwa akurasi tertinggi diperoleh saat jumlah partisi 12 yakni 97,56%. Semakin banyak jumlah partisi citra maka semakin baik akurasi yang didapatkan, dimana diperoleh *output* wajah yang mampu memetakan *input* ke target yang diharapkan dengan lebih tepat. Namun, jumlah partisi yang semakin banyak pada dasarnya membutuhkan lebih banyak waktu tahap pembelajaran, sehingga dibutuhkan waktu komputasi yang lebih lama.

3.3 Pengaruh Klasifikasi SVM Multiplikasi terhadap Performansi Sistem



Gambar 15. Grafik Hasil Pengujian Skenario 3



Gambar 16 Grafik Waktu Komputasi Skenario 3

Pada bagian ini, dilakukan pengujian sistem menggunakan skenario 3.

Berdasarkan Gambar 15 dan 16, diperoleh bahwa penggunaan parameter klasifikasi SVM multiplikasi OAA dan OAO memberikan akurasi pengujian yang berbeda. Setiap metode klasifikasi, khususnya SVM pada dasarnya membutuhkan ciri yang khas antar citra agar tidak terjadi kesalahan dalam melakukan pengenalan. Pada sistem ini, akurasi pengujian terbaik didapat saat penggunaan SVM multiplikasi SVM OAA dengan akurasi 97,5%.

4.5 Performansi Sistem

Berdasarkan pengujian pada skenario 1, 2, dan 3 diperoleh bahwa jenis iterasi, jumlah *point cloud* (partisi), dan SVM multiplikasi yang digunakan mempengaruhi akurasi pengujian. Pada bagian ini, akan dilakukan analisis akurasi pengujian sistem menggunakan parameter terbaik dari semua skenario.

Tabel 4 Akurasi Pengujian SVM OAA

Akurasi Pelatihan OAA	Akurasi Pengujian OAA	Waktu Komputasi (menit)
100,00%	97,56%	986,409

Dari Tabel 4 diperoleh pada iterasi hingga ke 49, partisi 12, dan SVM OAA adalah yang paling baik digunakan untuk mempresentasikan sebuah citra 3D sebagai dasar pengenalan. Pada kondisi tersebut, diperoleh akurasi pengujian terbaik yakni 97,56%. Berdasarkan pengujian dengan parameter terbaik yang telah dilakukan, diperoleh hasil perhitungan *Confusion Matrix*, sebagai berikut:

Tabel 5. *Confusion Matrix*

Nilai Prediksi	Nilai Sebenarnya	
	TRUE	FALSE
Nilai TRUE	TP = 36	FP = 1
Nilai FALSE	FN = 0	TN = 4

$$Precision = \frac{TP}{TP + FP} = \frac{36}{36 + 1} \times 100\% = 97,30\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{36}{36 + 0} \times 100\% = 100\%$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{36 + 4}{36 + 4 + 1 + 0} \times 100\% = 97,56\%$$

4. Kesimpulan

Implementasi *Iterative Closest Point* sebagai metode ekstraksi ciri dan *Support Vector Machine* sebagai metode klasifikasi telah mampu memberikan akurasi yang tinggi pada sistem identifikasi pengenalan wajah 3D, yakni mencapai 97,56% dengan waktu komputasi 986,409 menit/80 citra uji. Akurasi tersebut diperoleh saat nilai dari masing-masing parameter sistem adalah sebagai berikut: jumlah iterasi=49, jumlah *point cloud*/partisi=12. Sistem ini bekerja dengan baik pada penggunaan klasifikasi SVM Multiplikasi *One-Against-All* (OAA). Pada iterasi ke 49, partisi 12, dan

menggunakan SVM OAA, diketahui bahwa sistem ini bekerja optimal dengan *precision* 97,30%, *recall* 100,00% dan *accuracy* 97,56%.

DAFTAR PUSTAKA

- ABATE, A.F., NAPPI, M., RICCIO, D., SABATINO, G., 2007. 2D and 3D face recognition: A survey, *Pattern Recognition Letters*.
- ABDELMOTTLEP, M.A., 2016. World Internal Security & Police Index, International Science Association (IPSA), Florida.
- AJMERA, R., NIGAM, A., GUPTA, P, 2014. 3D Face Recognition using Kinect, *ICVGIP'14 Proceedings of the Indian Conference on Conference on Computer Vision Graphics and Image Processing Article No.76*, India.
- AFRILIANA, I, 2015. Pengolahan Citra Digital, Poltek Harapan Bangsa.
- ALABBASI, H.A., MOLDOVEANU, F., 2015. Real Time Facial Emotion Recognition using Kinect V2.0 Sensor, *IOSR Journal of Computer Engineering (IOSR-JCE)* Volum 17, Issue 3, Ver. II.
- AMOR, B.B., OUJI, K., ARDABILIAN, M., CHEN. L., 2008. 3D Face recognition by ICP-based shape matching” *LIRIS Lab, Lyon Research Center for Images and Intelligent Information System*.
- BASIR, B., WARDI, ZAINUDDIN, Z., Agustus 2017. Sistem Keamanan Rumah Berbasis Kinect. *Jurnal IT*, Volume 8 No 2. 2017.
- BUDI, A. SUMATINNA, S., MAULANA, H., 2016. Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA), *Jurnal Teknik Informatika Vol 9 No. 2*.
- COOK, J. 2004. Face Recognition from 3D Data using Iterative Closest Point Algorithm and Gaussian Mixture Models. *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04)*.
- DHRITI, KAUR, M., 2012. K-Nearest Neighbor Classification Approach for Face and Fingerprint at Feature Level Fusion. *International Journal of Computer Applications (0975 – 8887)*, Volume 60– No.14.
- FRANK, B. 2008. A 3D Face Matching Framework. *IEEE International Conference on Shape Modeling and Applications*.
- GOOS, J.H.A.J.V.L.G., 2002. *Pattern Recognition with Support Vector Machines*. NewYork: Springer.
- HAAR, F.B., VELTKAMP, R.C., 2009. A 3D Face Matching Framework, Department of Information and Computing Sciences, Utrecht University, Netherland.
- HIREMATH, M., HIREMATH, P.S., 2017. 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*.
- LI, B., MIAN, A., LIU, W., & KRISHNA, A. 2013. Using kinect for face recognition under varying poses, expressions, illumination and disguise, in *Applications of Computer Vision (WACV) IEEE Workshop on*, pp. 186–192.
- PRASETYO, E., 2014. *Data Mining-Mengolah Data Menjadi Informasi Menggunakan Matlab*, Yogyakarta.
- MUJIB, K., HIDAYATNO, A., PRAKOSO, T., 2018. Pengenalan Wajah Menggunakan Local Binary Pattern (LBP) dan Support Vector Machine (SVM),” *TRANSIENT*, vol. 7, no. 1, pp. 124-130.
- MRACEK, S., DRAHANSK, M., DVORAK, R., PROVAZNIK, I., VANA, J., 2014. 3D Face Recognition on Low-Cost Depth Sensors.
- NURAJIZAH, S., 2013. Penerapan Metode Support Vector Machine Berbasis Particle. *Jurnal Techno Nusa Mandiri*, vol. 1, no. 1, pp. 216-226.
- PRASETYO, E., 2014. *Data Mining-Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta.
- PRAWESTININGTYAS, A.M.A.E., 2009. Forensic Identification Based on Both Primary and Secondary Examination Priority in Victim Identifiers on Two Different Mass Disaster Cases, *Jurnal Kedokteran Brawijaya*, Vol XXV, No.2, p. 87.
- SADA, I.H., TRITOASMORO, I.I., BUDIMAN, G., 2011. Pemodelan Wajah 3D Melalui Pendeteksian Fitur Wajah 2D Menggunakan Teknik Morphing,” *Universitas Telkom*.
- THOME, A.C.G., 2012. *SVM Classifiers – Concepts and Applications to Character Recognition*. Federal University of Rio de Janeiro, Brasil November 7th.
- WIDODO, P.P., HANDAYANTO, R.T., HERLAWATI, 2013. *Penerapan Data Mining Dengan Matlab*. Bandung: Rekayasa Sains.
- YANG, H. 2015. An Improved Iterative Closest Points Algorithm. *World Journal of Engineering and Technology*, 3, 302-308.
- ZHANG, D., & LU, G., 2012. *3D Biometrics Systems and Applications*, Hong Kong: Springer.

Halaman ini sengaja dikosongkan