

Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor

Fandiansyah¹, Jayanti Yusmah Sari², Ika Purwanti Ningrum³,
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari, 93232, Indonesia
lastfandiansyah@gmail.com¹, jayanti.yusmah.sari@gmail.com², ika.purwanti.n@gmail.com³

Diterima 12 April 2017

Disetujui 2 Juni 2017

Abstract—Face recognition is one of the biometric system that mostly used for individual recognition in the absent machine or access control. This is because the face is the most visible part of human anatomy and serves as the first distinguishing factor of a human being. Feature extraction and classification are the key to face recognition, as they are to any pattern classification task.

In this paper, we describe a face recognition method based on Linear Discriminant Analysis (LDA) and k -Nearest Neighbor classifier. LDA used for feature extraction, which directly extracts the proper features from image matrices with the objective of maximizing between-class variations and minimizing within-class variations. The features of a testing image will be compared to the features of database image using k -Nearest Neighbor classifier. The experiments in this paper are performed by using using 66 face images of 22 different people. The experimental result shows that the recognition accuracy is up to 98.33%.

Index Terms—face recognition, k nearest neighbor, linear discriminant analysis.

I. PENDAHULUAN

Sistem biometrika adalah teknologi pengenalan diri menggunakan bagian tubuh manusia seperti sidik jari, telinga, wajah, geometri tangan, telapak tangan, retina, gigi dan bibir [1]. Pengenalan wajah menjadi salah satu sistem biomterika yang paling banyak digunakan untuk identifikasi personal, misalnya pada penggunaan mesin absensi atau akses *control*. Hal ini karena wajah merupakan salah satu biometrika yang paling umum digunakan untuk mengenali seseorang [2]. Selain itu, pengenalan wajah juga tidak mengganggu kenyamanan seseorang saat akuisisi citra. Pada sistem pengenalan wajah, citra wajah dari seorang individu akan diakuisisi sebagai citra masukan yang kemudian dicocokkan dengan sekumpulan citra wajah yang sebelumnya telah disimpan di dalam *database* untuk mengenali citra wajah dari individu tersebut.

Penelitian tentang pengenalan wajah telah dimulai sejak tahun 1970-an [3]. Secara umum, ada 4 tahapan

dalam sistem pengenalan wajah yaitu akuisisi citra wajah, *preprocessing*, ekstraksi fitur dan klasifikasi citra wajah tersebut. Dari keempat tahapan tersebut, tahapan ekstraksi fitur dan klasifikasi merupakan tahapan yang paling penting dalam sistem pengenalan wajah [4].

Ada banyak metode ekstraksi fitur yang telah dikembangkan untuk membangun sistem pengenalan wajah. Dari semua metode tersebut, masing-masing memiliki kelebihan dan kelemahan. Ada metode yang membutuhkan waktu komputasi yang cepat namun mengorbankan tingkat akurasi pengenalan, dan ada pula metode dengan tingkat akurasi pengenalan yang tinggi namun membutuhkan waktu komputasi yang lama. Baik waktu komputasi maupun tingkat akurasi pada sistem pengenalan wajah, keduanya dipengaruhi oleh metode yang digunakan pada tahap ekstraksi fitur [4] dan metode yang digunakan pada tahap klasifikasi [5]. Oleh karena itu, untuk mengoptimalkan waktu komputasi tanpa harus mengorbankan tingkat akurasi pengenalan, penelitian ini membangun sistem pengenalan wajah dengan memfokuskan pada kedua tahapan yaitu tahapan ekstraksi fitur dan tahapan klasifikasi.

Salah satu metode ekstraksi fitur yang telah digunakan untuk sistem pengenalan wajah yaitu *Principal Component Analysis* (PCA). Metode PCA bertujuan untuk mereduksi dimensi dengan melakukan transformasi linear dari suatu ruang berdimensi tinggi ke dalam ruang berdimensi rendah [6]. Namun, kelemahan dari metode PCA adalah kurang optimal dalam pemisahan antar kelas sehingga dapat mempengaruhi tingkat akurasi pengenalan wajah [7]. Untuk mengatasi masalah tersebut, Belhumeur dkk [7] memperkenalkan *Linear Discriminant Analysis* (LDA) untuk pengenalan wajah. Metode ini bekerja dengan menemukan subruang linear yang memaksimalkan perpisahan dua kelas pola menurut *Fisher Criterion* (bobot *fisher* kriteria). Hal ini dapat diperoleh dengan

meminimalkan jarak matriks sebaran *within-class* S_w dan memaksimalkan jarak matriks sebaran *between-class* S_b secara simultan sehingga menghasilkan *fisher criterion* yang maksimal. LDA akan menemukan subruang linear di mana kelas-kelas saling terpisah dengan memaksimalkan *fisher criterion* [2]. Metode LDA digunakan dalam penelitian ini sebagai metode ekstraksi fitur untuk mengoptimalkan pemisahan antar kelas citra wajah sehingga hasil pengenalannya lebih akurat. Dengan demikian, metode LDA akan diuji berdasarkan tingkat akurasi pengenalan.

Setelah melalui tahapan ekstraksi fitur, fitur-fitur penting dari citra wajah nantinya akan digunakan untuk proses klasifikasi atau pengenalan. Metode klasifikasi yang akan digunakan dalam penelitian ini adalah metode klasifikasi *k nearest neighbor*. Metode klasifikasi *k nearest neighbor* memiliki komputasi yang lebih sederhana namun kemampuan klasifikasinya lebih baik dibandingkan dengan metode lain seperti Hidden Markov Model dan metode Kernel method [5]. Metode *k nearest neighbor* adalah suatu metode yang menggunakan algoritma *supervised* di mana hasil dari *query instance* masukan akan diklasifikasikan berdasarkan mayoritas dari kategori pada *k nearest neighbor* [5]. Penelitian ini bertujuan untuk mengimplementasikan metode ekstraksi fitur *Linear Discriminant Analysis* (LDA) dan metode klasifikasi *k nearest neighbor* untuk membangun sistem pengenalan wajah dengan tingkat akurasi pengenalan yang tinggi dan waktu komputasi yang optimal.

Pembahasan penelitian ini dibagi dalam beberapa bagian. Analisis masalah dan penelitian terkait diuraikan pada bagian "PENDAHUULUAN". Metodologi penelitian yang diusulkan dibahas pada bagian "METODE PENELITIAN". Selanjutnya, hasil pengujian dibahas pada bagian "HASIL dan PEMBAHASAN". Adapun kesimpulan dan saran untuk penelitian selanjutnya dibahas pada bagian "KESIMPULAN dan SARAN".

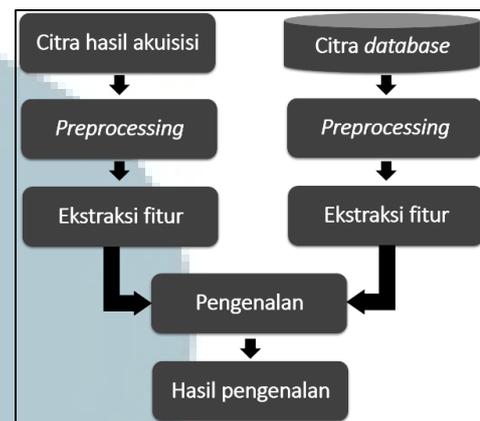
II. METODE PENELITIAN

A. Perancangan Sistem

Sistem pengenalan wajah menggunakan metode *Linear Discriminant Analysis* (LDA) dan *k nearest neighbor* dirancang agar dapat mendeteksi wajah seseorang pada citra digital kemudian mengenali citra wajah tersebut dengan cara mencocokkan hasil ekstraksi fiturnya dengan fitur citra wajah yang sudah disimpan di dalam *database*.

Dalam penelitian ini, digunakan 66 citra wajah yang berformat *.png dan berasal dari 22 Mahasiswa Jurusan Teknik Informatika Universitas Halu Oleo.

Setiap individu masing-masing mempunyai 3 citra wajah dengan ekspresi yang berbeda-beda. Jumlah data citra wajah yang akan diidentifikasi sebanyak 56 citra meliputi: 20 citra dari 20 individu yang datanya ada di *database*, 20 citra dari 20 individu yang datanya ada di *database* namun diberikan gangguan *ocean distortion*, 12 citra dari 12 individu yang datanya ada di dalam *database* namun diakuisisi dengan pose wajah yang berbeda, serta 4 citra dari 4 individu yang berbeda namun diakuisisi dengan pencahayaan yang berbeda. Gambar. 1 menunjukkan gambaran umum sistem yang diusulkan.



Gambar 1. Gambaran Umum Sistem

B. Perancangan Proses

Perancangan proses dibagi menjadi dua yaitu perancangan tahap *preprocessing*, dan perancangan tahap *processing* yang meliputi ekstraksi fitur dan pengenalan (*recognize*). Gambar 1 menggambarkan proses yang dilakukan oleh sistem. Proses yang akan dilakukan pertama adalah tahap akuisisi citra *testing* maupun citra *database* berfungsi untuk pengambilan citra wajah, tahap *preprocessing* yang berfungsi untuk menyeragamkan ukuran citra wajah dan memperbaiki kualitasnya sebelum melalui tahap ekstraksi fitur. Selanjutnya akan dimulai pembentukan fitur citra *testing* yang berfungsi mengekstraksi ciri fitur citra untuk dipergunakan sebagai perhitungan mencari bobot.

Tahap pembentukan fitur citra *database* berfungsi sama seperti ekstraksi citra *testing*. Proses pengenalan menggunakan metode klasifikasi *k nearest neighbor* yang berfungsi untuk mengelompokkan informasi fitur citra dari proses pembentukan fitur citra *database*, kemudian dilakukan klasifikasi pada sistem dengan data yang sudah diberikan *label*.

C. Preprocessing citra

Citra wajah yang disimpan di *database* dan citra yang akan diidentifikasi harus melalui tahap

preprocessing terlebih dahulu yang meliputi akuisisi citra, konversi citra RGB-*grayscale*, dan ekualisasi *histogram*. Berikut tahapan *preprocessing* yang harus dilakukan secara berurutan.

- 1) Akuisisi citra wajah menggunakan *webcam* laptop. Citra hasil akuisisi merupakan citra RGB 24 bit berformat PNG dengan ukuran 92 x 112 piksel.
- 2) Konversi citra wajah hasil akuisisi dari RGB ke *grayscale*.
- 3) Ekualisasi *histogram (histogram equalization)*, hasilnya akan disimpan untuk proses pengenalan selanjutnya yang terdiri dari 66 citra yang ada di dalam *database* wajah dan 22 citra sebagai citra *testing*. Tahapan ini merupakan tahapan terakhir dari *preprocessing* citra wajah.

D. Processing Citra

Pada tahapan *processing*, akan diterapkan metode LDA untuk menghasilkan vektor fitur dari citra wajah dan melakukan pencocokan vektor fitur citra di *database* dengan vektor fitur citra *testing* menggunakan perhitungan *k nearest neighbor*.

D.1. Proses Pembentukan Vektor Fitur Citra

Setelah melakukan tahapan *preprocessing*, maka selanjutnya adalah proses ekstraksi fitur citra. Ekstraksi fitur dilakukan pada citra *testing* maupun citra yang ada di dalam *database*. Pada tahap awal proses ekstraksi fitur pada penelitian ini digunakan Algoritma *Eigenface* sebagai salah satu algoritma berbasis metode PCA (*Principal Component Analysis*) untuk memperoleh bobot. Bobot tersebut selanjutnya akan digunakan pada metode LDA (*Linear Discriminant Analysis*) untuk membentuk vektor fitur citra wajah.

Perhitungan PCA [6]

Eigenface merupakan salah satu algoritma pengenalan wajah manusia yang berbasis metode PCA. Untuk menghasilkan *eigenface*, sekumpulan citra digital dari wajah manusia diambil pada kondisi pencahayaan yang sama kemudian dinormalisasikan dan diproses pada resolusi yang sama (misal $m \times n$), kemudian citra tersebut diperlakukan sebagai vektor dimensi $m \times n$ di mana komponennya diambil dari nilai piksel citra.

Berikut langkah-langkah Algoritma *Eigenface* untuk memperoleh bobot dari citra *database*.

- a. Mengubah M jumlah citra di *database* dengan ukuran $B \times K$ (92 x 112) piksel ke dalam bentuk vektor Y dengan panjang 1×10304 .

$$X = \begin{bmatrix} U_{1,1} & \dots & U_{1,92} \\ \vdots & \ddots & \vdots \\ U_{112,1} & \dots & U_{112,92} \end{bmatrix} \quad (1)$$

$$Y = [\Gamma_1, \Gamma_2, \dots, \dots, \Gamma_{10304}] \quad (2)$$

- b. Simpan vektor dari 66 citra di *database* ke dalam *list image* (S).

$$S = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \dots \\ \Gamma_{66} \end{bmatrix} = \begin{bmatrix} U_{1,1} & U_{1,2} & \dots & U_{1,10304} & 1 \\ U_{2,1} & U_{2,2} & \dots & U_{2,10304} & \\ \dots & \dots & \dots & \dots & \\ U_{66,1} & U_{66,2} & \dots & U_{66,10304} & \end{bmatrix} \quad (3)$$

- c. Mencari matriks rata-rata (Ψ) dari semua piksel citra di *database* P_m dengan m merupakan jumlah piksel citra dan M adalah jumlah citra di *database*, dengan menggunakan persamaan (4) sehingga dihasilkan matriks vektor rata-rata dengan dimensi 1×10304 .

$$\Psi = \frac{1}{M} \sum_{m=1}^M P_m ; m = 1, 2, \dots, M \quad (4)$$

- d. Menghitung selisih (Φ) antara citra di *database* Γ_m dengan nilai tengah (Ψ) menggunakan persamaan (5) sehingga diperoleh matriks selisih Φ berukuran 66×10304 .

$$\Phi = \Gamma_m - \Psi \quad (5)$$

- e. Kemudian tentukan matriks kovarian C dengan mengalikan selisih Φ dan matriks *transpose* Φ^T menggunakan persamaan (6). Matriks C yang diperoleh nanti berukuran 66×66 .

$$C = \Phi \Phi^T \quad (6)$$

- f. Tentukan nilai *eigenvalue* (λ) dan *eigenvector* (v) dari matriks kovarian C menggunakan persamaan (7). (λ) yang diperoleh berukuran 1×66 dan sedangkan v berukuran 66×66 .

$$\begin{aligned} C V &= \lambda V \\ (C - \lambda I) V &= 0 \\ (\Phi \Phi^T - \lambda I) V &= 0 \end{aligned} \quad (7)$$

Dengan I adalah matriks identitas, λ adalah *eigenvalue* dari C dan V adalah *eigenvector* yang bersesuaian dengan λ . Untuk mencari nilai λ , maka *determinant* ($C - \lambda I$) harus sama dengan 0. Setiap citra akan memiliki n buah λ , ambil λ dengan nilai tertinggi dan cari *eigenvector* yang bersesuaian dengan λ tersebut. Setiap *eigenvector* akan memiliki panjang M elemen.

- g. Masukkan nilai *eigenvector* (v) yang dipilih ke dalam matriks *eigenface* (E_k) dengan mengalikan v yang berukuran $M \times M$ ($M=66$) dengan matriks selisih Φ yang berukuran $M \times BK$ ($BK=10304$) dan simpan hasilnya dalam matriks yang berukuran $M \times BK$.

$$E_k = \sum_{k=1}^M v_k \Phi_k ; k = 1, 2, \dots, M \quad (8)$$

- h. Normalisasi *eigenface* dengan menggunakan persamaan (9), dan simpan *eigenface* hasil

normalisasi ini ke dalam matriks E_n berukuran 45×10304 pada *feature extraction*.

$$E_{nk} = \frac{E_k}{E_k, E_k^T} \quad (9)$$

- i. Hitung bobot dengan mengalikan matriks selis Φ_k berukuran $M \times BK$ dengan matriks *transpose* dari E_{nk} yang berukuran $BK \times M$, menggunakan persamaan (10).

$$w = \Phi_k \times E_{nk}^T \quad (10)$$

- j. Bobot yang diperoleh merupakan nilai ciri yang yang digunakan dalam proses pengenalan selanjutnya. Simpan bobot ini ke dalam matriks Ω berukuran $M \times N$ dengan $N < M$.

Perhitungan LDA

LDA bekerja berdasarkan analisis matriks penyebaran yang bertujuan menemukan suatu proyeksi optimal sehingga dapat memproyeksikan data *input* pada ruang dengan dimensi yang lebih kecil dimana semua pola dapat dipisahkan semaksimal mungkin. Karenanya untuk tujuan pemisahan tersebut maka LDA akan mencoba untuk memaksimalkan penyebaran data-data *input* di antara kelas-kelas yang berbeda dan sekaligus juga meminimalkan penyebaran *input* pada kelas yang sama. Perbedaan antar kelas direpresentasikan oleh matriks S_b dan perbedaan dalam kelas direpresentasikan oleh matriks S_w . Berikut langkah-langkah pembentukan vektor fitur citra wajah menggunakan metode LDA [8][9].

- a. Hasil dari bobot *eigenfaces* (E_k) dijadikan sebagai input yang akan ditranformasikan ke dalam vektor kolom.

- b. Menghitung rata-rata dalam kelas (m_i) dan rata-rata keseluruhan kelas (m) dari seluruh citra di *database*.

- c. Menghitung matriks sebaran antar kelas (*between class scatter matrix*, S_b).

$$S_b = \sum_{i=1}^k n_i (m_i - m_0)(m_i - m_0)^T \quad (11)$$

- d. Menghitung matriks sebaran dalam kelas (*within class scatter matrix* S_w).

$$S_w = \sum_{i=1}^k \sum_{j=1}^{n_i} (x_i^{(j)} - m_i)(x_i^{(j)} - m_i)^T \quad (12)$$

- e. Memproyeksikan matriks sebaran dalam kelas. Matriks sebaran dalam kelas (S_w) adalah jarak matriks dalam kelas yang sama, menggunakan persamaan (13).

$$J_2(W) = \max \text{trace} ((W^T S_w W)^{-1} (W^T S_b W)) \quad (13)$$

- f. Mencari *eigen value* (λ) dan nilai *eigenvector* (v) menggunakan persamaan (14).

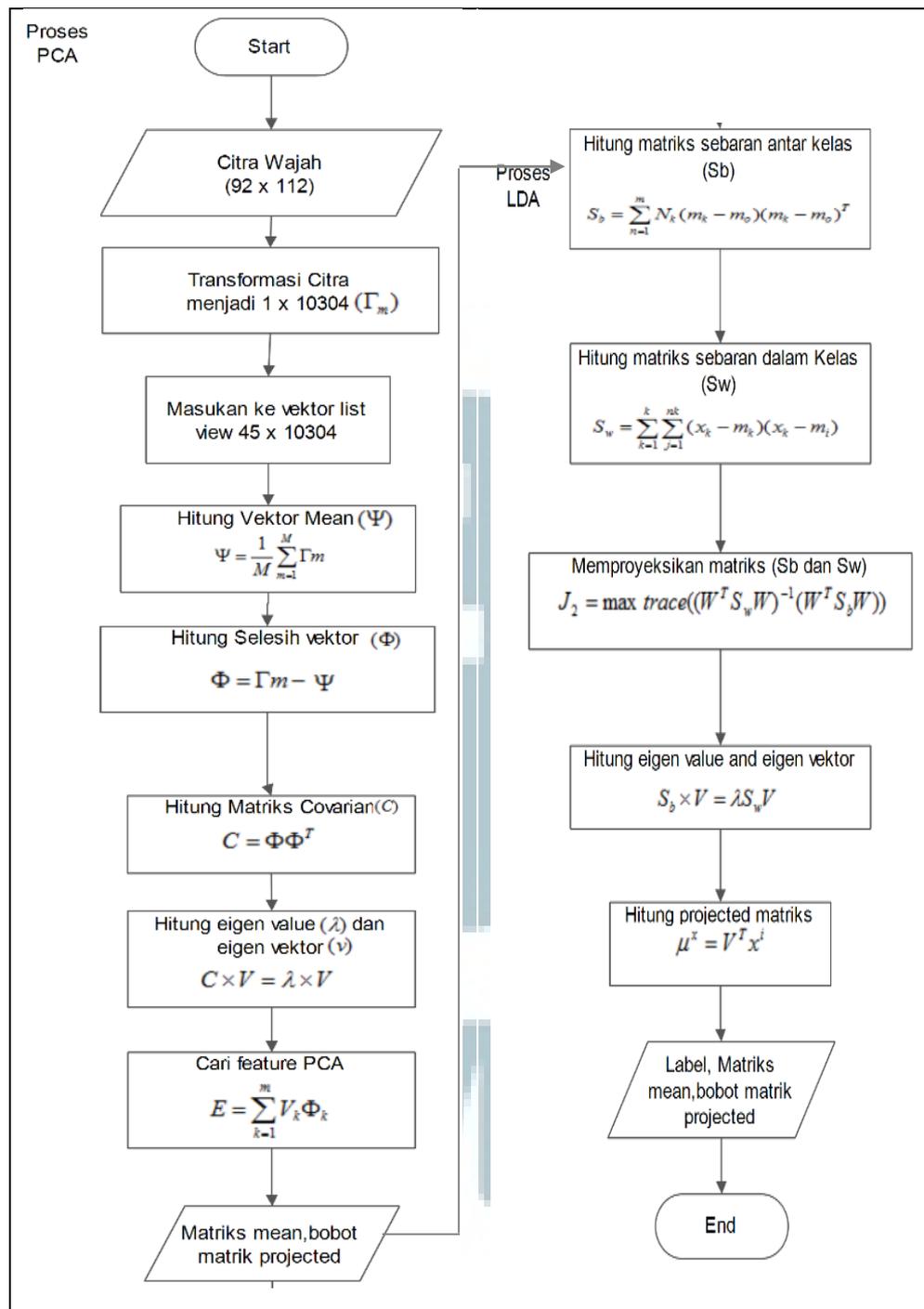
$$S_b v = \lambda S_w v \quad (14)$$

- g. Mengurutkan *eigen value* (λ) sesuai dengan urutan nilai yang ada pada nilai eigen dari besar ke kecil. Selanjutnya proyeksi menggunakan $k-1$ *eigenvector* (v) (di mana k adalah jumlah kelas).

- h. Memproyeksikan seluruh citra asal (bukan *centered image*) ke *fisher basis* vektor dengan menghitung *dot product* dari citra asal V^T ke tiap *fisher basis vector* x^i menggunakan persamaan (15).

$$\bar{u}^x = V^T x^i \quad (15)$$

UMN

Gambar 2. Proses Pembentukan Vektor Fitur Citra *Database*

Pembentukan vektor fitur untuk pengenalan wajah memiliki beberapa tahapan seperti yang telah ditunjukkan pada Gambar 2. Untuk ekstraksi fitur citra *testing* yang telah melalui tahap *preprocessing* dilakukan langkah-langkah yang sama dengan ekstraksi fitur citra pada citra *database*. Perbedaannya

hanya terletak pada jumlah citra (M), jika pada ekstraksi fitur citra *database* nilai $M=66$, maka pada ekstraksi fitur citra *testing*, nilai $M=1$.

D.2. Proses Pencocokan Fitur

Pencocokan bertujuan untuk menentukan kelas dari suatu citra *testing* berdasarkan ciri-ciri yang telah diekstraksi. Proses pencocokan fitur hasil ekstraksi menggunakan metode klasifikasi *k nearest neighbor* (*kNN*). Metode klasifikasi *k nearest neighbor* merupakan pengembangan dari metode klasifikasi *nearest neighbor* (*NN*) [5]. Pada metode klasifikasi *nearest neighbor*, citra masukan akan diuji berdasarkan jarak fiturnya dengan fitur citra lain di dalam *database* untuk memperoleh satu citra dengan nilai jarak fitur paling minimum. Citra ini disebut dengan ketetangaan terdekat.

Pada metode klasifikasi *nearest neighbor*, diperoleh satu citra yang memiliki jarak minimum dengan citra uji. Sedangkan pada metode klasifikasi *k nearest neighbor* dihasilkan citra sample sejumlah *k* yang memiliki jarak minimum dengan citra uji. Setiap citra sampel tersebut masing-masing memiliki label kelas. Citra uji akan diklasifikasikan ke dalam suatu kelas dengan jumlah citra sampel yang paling banyak dari citra sample sejumlah *k*. Nilai *k* pada *kNN* ini mempengaruhi performansi dari metode klasifikasi *kNN* [5].

Metode klasifikasi *k nearest neighbor* melakukan proses pencocokan/pengenalan berdasarkan jumlah tetangga terdekat untuk penentuan kelasnya. Untuk mencari jarak kelas menggunakan perhitungan jarak *euclidean distance*. Tahapan dalam metode klasifikasi *k nearest neighbor* yaitu [10]:

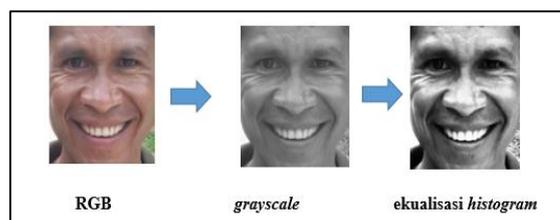
- Menentukan nilai *k*.
- Menghitung jarak antara citra *testing* dengan seluruh citra pada *database* menggunakan persamaan *euclidean distance*, persamaan (16) dan menentukan citra terdekat dengan citra *testing* berdasarkan nilai *k*.
- Menentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak.
- Jika terjadi konflik atau keadaan seimbang pada kelas dengan jumlah anggota yang sama maka digunakan pemecahan konflik.

III. HASIL DAN PEMBAHASAN

A. Preprocessing

Tahap *preprocessing* berfungsi untuk menyiapkan citra sebelum proses ekstraksi fitur. *Preprocessing* dalam penelitian ini meliputi akuisisi citra, konversi citra RGB-*grayscale* dan ekualisasi *histogram*. Dari Gambar 3, dapat dilihat bahwa *output* dari tahap *preprocessing* berupa citra

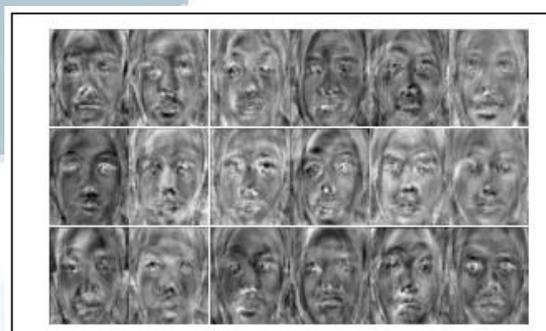
grayscale yang telah mengalami ekualisasi *histogram*.



Gambar 3. Contoh Hasil *Preprocessing*

B. Ekstraksi fitur

Selanjutnya citra akan diproses melalui tahapan ekstraksi fitur untuk membuat suatu *set fisherface* dari citra *database* menggunakan perhitungan *Principal Component Analysis* dan *Linear Discriminant Analysis*. *Fisherface* ini sebenarnya merupakan seatur vektor eigen dengan nilai eigen tertentu yang ditampilkan ke dalam gambar dua dimensi dengan *mode grayscale*. Suatu *set fisherface* setiap gambar wajah dapat diekstraksi kembali dengan bobot yang berbeda beda dari tiap *fisherface*. Bobot *fisherface* tersebut dibandingkan untuk melakukan pencocokan dari citra *testing*.



Gambar 4. Hasil Ekstraksi Fitur Citra *Database*

Gambar 4 merupakan hasil dari ekstraksi fitur citra *database* menggunakan metode LDA, walaupun hasil ekstraksi fitur tidak menyerupai gambar aslinya, akan tetapi setiap gambar yang merupakan satu kelas memiliki gambar yang hampir sama yang diproyeksikan ke dalam kelas yang sama perbedaan gambarnya dapat dihilangkan atau diekstraksi dengan menjadi satu gambar.

C. Hasil Pengujian

Pengujian sistem pengenalan wajah menggunakan metode LDA dan *k-NN* pada penelitian ini dilakukan dengan dua skenario uji coba. Pengujian dilakukan dengan menggunakan 20 citra wajah dari individu yang citra wajahnya sudah

disimpan terlebih dahulu di dalam *database*. Setiap individu memiliki 3 sampel citra wajah pada *database*.

Pengujian I menggunakan citra uji normal, yaitu citra wajah hasil akuisisi dengan *pose* dan ekspresi berbeda-beda. Hasil dari pengujian ini ditunjukkan pada Tabel 1. Adapun pengujian II dilakukan menggunakan citra uji dengan gangguan *ocean distortion*. Pengujian ini bertujuan untuk mengukur performansi sistem pengenalan wajah yang dibuat dalam penelitian ini. Hasil dari pengujian kedua ditunjukkan dalam Tabel 2. Hasil pengujian pada Tabel 1 dan Tabel 2 berupa pengenalan yang kurang

tepat yaitu citra uji dikenali sebagai wajah dari individu lain, diberi tanda kotak merah.

Tabel 1 menunjukkan bahwa dari hasil uji coba pertama, yaitu pengujian dengan citra wajah normal menggunakan nilai $k=3, 5$ dan 7 , diperoleh satu pengenalan wajah yang kurang tepat yaitu pada individu ke-12. Sedangkan pada Tabel 2, dapat dilihat bahwa terdapat 8 kesalahan pengenalan saat uji coba kedua menggunakan citra wajah yang diberi efek gangguan *ocean distortion* yaitu pada individu ke-6 dan ke-12 untuk $k=3$, pada individu ke-11, 12 dan 13 untuk $k=5$ dan pada individu ke-9, 12 dan 13 untuk $k=7$.

Tabel 1 Hasil Pengujian I Menggunakan Citra Uji Normal

No	Citra Testing	Hasil pengenalan menggunakan <i>kNN</i> disertai jarak (d)			No	Citra Testing	Hasil pengenalan menggunakan <i>kNN</i>		
		$k=3$	$k=5$	$k=7$			$k=3$	$k=5$	$k=7$
1		 $d = 0.1798$	 $d = 0.1485$	 $d = 0.1485$	11		 $d = 0.1054$	 $d = 0.10548$	 $d = 0.1305$
2		 $d = 0.2705$	 $d = 0.2529$	 $d = 0.2323$	12		 $d = 0.1942$	 $d = 0.1942$	 $d = 0.2983$
3		 $d = 0.0638$	 $d = 0.1014$	 $d = 0.2577$	13		 $d = 0.2035$	 $d = 0.3035$	 $d = 0.3894$
4		 $d = 0.0930$	 $d = 0.1184$	 $d = 0.1474$	14		 $d = 0.1899$	 $d = 0.1903$	 $d = 0.4398$
5		 $d = 0.3567$	 $d = 0.3238$	 $d = 0.3238$	15		 $d = 0.2866$	 $d = 0.2886$	 $d = 0.3429$
6		 $d = 0.2853$	 $d = 0.3233$	 $d = 0.3238$	16		 $d = 0.1987$	 $d = 0.2289$	 $d = 0.2389$
7		 $d = 0.0972$	 $d = 0.0659$	 $d = 0.0955$	17		 $d = 0.1142$	 $d = 0.1873$	 $d = 0.1873$
8		 $d = 0.1939$	 $d = 0.2199$	 $d = 0.2199$	18		 $d = 0.1639$	 $d = 0.2254$	 $d = 0.2254$
9		 $d = 0.2388$	 $d = 0.2493$	 $d = 0.1762$	19		 $d = 0.1375$	 $d = 0.1459$	 $d = 0.7834$
10		 $d = 0.2707$	 $d = 0.3060$	 $d = 0.0760$	20		 $d = 0.2108$	 $d = 0.2108$	 $d = 0.2108$

Tabel 2 Hasil Pengujian II Menggunakan Citra Uji Dengan Penambahan Gangguan *Ocean Distortion*

No	Citra <i>Testing</i>	Hasil pengenalan menggunakan <i>kNN</i> disertai jarak (<i>d</i>)			No	Citra <i>Testing</i>	Hasil pengenalan menggunakan <i>kNN</i>		
		<i>k</i> =3	<i>k</i> =5	<i>k</i> =7			<i>k</i> =3	<i>k</i> =5	<i>k</i> =7
1		 <i>d</i> = 0.0881	 <i>l</i> = 0.1177	 <i>l</i> = 0.1177	11		 <i>l</i> = 0.1320	 <i>l</i> = 0.1320	 <i>l</i> = 0.1320
2		 <i>d</i> = 0.1267	 <i>l</i> = 0.1267	 <i>l</i> = 0.1267	12		 <i>l</i> = 0.0319	 <i>l</i> = 0.0627	 <i>l</i> = 0.0627
3		 <i>d</i> = 0.0760	 <i>l</i> = 0.1091	 <i>l</i> = 0.1091	13		 <i>l</i> = 0.0940	 <i>l</i> = 0.1203	 <i>l</i> = 0.1203
4		 <i>d</i> = 0.1626	 <i>l</i> = 0.1626	 <i>l</i> = 0.1626	14		 <i>l</i> = 0.1824	 <i>l</i> = 0.1824	 <i>l</i> = 0.1824
5		 <i>d</i> = 0.1672	 <i>l</i> = 0.1672	 <i>l</i> = 0.1672	15		 <i>l</i> = 0.1732	 <i>l</i> = 0.1752	 <i>l</i> = 0.2939
6		 <i>d</i> = 0.1427	 <i>l</i> = 0.1784	 <i>l</i> = 0.1784	16		 <i>l</i> = 0.1767	 <i>l</i> = 0.1767	 <i>l</i> = 0.1912
7		 <i>d</i> = 0.1677	 <i>l</i> = 0.1677	 <i>l</i> = 0.1677	17		 <i>l</i> = 0.1253	 <i>l</i> = 0.1284	 <i>l</i> = 0.1252
8		 <i>d</i> = 0.2297	 <i>l</i> = 0.1097	 <i>l</i> = 0.2297	18		 <i>l</i> = 0.1808	 <i>l</i> = 0.1808	 <i>l</i> = 0.1802
9		 <i>d</i> = 0.2218	 <i>l</i> = 0.2218	 <i>l</i> = 0.2218	19		 <i>l</i> = 0.1101	 <i>l</i> = 0.1459	 <i>l</i> = 0.1101
10		 <i>d</i> = 0.1380	 <i>l</i> = 0.1380	 <i>l</i> = 0.1380	20		 <i>l</i> = 0.1398	 <i>l</i> = 0.1402	 <i>l</i> = 0.1398

Tabel 3. Hasil Pengujian I

Nilai <i>k</i>	Benar	Salah	Citra <i>Testing</i>	Akurasi
3	20	0	20	100%
5	20	0	20	100%
7	19	1	20	95%
Rata-rata akurasi pengenalan				98.33%

Tabel 4. Hasil Pengujian II

Nilai <i>k</i>	Benar	Salah	Citra <i>Testing</i>	Akurasi
3	18	2	20	90%
5	17	3	20	85%
7	17	3	20	85%
Rata-rata akurasi pengenalan				86.66%

Tabel 3 dan Tabel 4 menunjukkan akurasi pengenalan terhadap citra uji yang normal dan citra uji dengan gangguan *ocean distortion*. Akurasi pengenalan dihitung berdasarkan hasil pengujian

pada Tabel 1 dan 2, dengan membandingkan jumlah citra uji dengan hasil pengenalan benar terhadap jumlah seluruh citra yang diuji dengan menggunakan persamaan (17).

$$Akurasi = \frac{\text{Total Pengenalan benar}}{\text{Total Citra Uji}} \times 100 \% \quad (17)$$

Dari Tabel 3 dapat dilihat bahwa nilai akurasi tertinggi yaitu 100% untuk pengujian pertama, yaitu pengenalan citra uji normal diperoleh dengan $k=3$ dan $k=5$. Sedangkan untuk nilai akurasi tertinggi yaitu 90% pada pengujian kedua seperti yang ditunjukkan pada Tabel 4, diperoleh pada nilai $k=3$. Hal ini tidak berarti bahwa nilai $k=3$ lebih baik dibandingkan dengan $k=5$ atau 7. Hasil uji coba ini hanya menunjukkan bahwa nilai k yang sesuai dengan data pada penelitian ini adalah $k=3$. Adapun akurasi rata-rata yang diperoleh pada pengujian pertama dan kedua secara berturut-turut yaitu sebesar 98.33% dan 86.66%. Hasil pengujian ini membuktikan bahwa metode *Linear Discriminant Analysis* dan *k nearest neighbor* mampu mengoptimalkan hasil pengenalan pada sistem pengenalan wajah. Hal ini terlihat dari tingkat akurasi pengenalan yang cukup tinggi baik pada pengujian I yang menggunakan citra uji normal maupun pada pengujian II yang menggunakan citra uji dengan gangguan *ocean distortion*.

IV. KESIMPULAN

Dalam penelitian ini, dibangun sistem pengenalan wajah menggunakan metode *Linear Discriminant Analysis* dan *k nearest neighbor*. Metode *Linear Discriminant Analysis* digunakan untuk mengekstraksi fitur citra wajah sedangkan metode *k nearest neighbor* digunakan untuk mencocokkan fitur citra wajah pada tahap pengenalan. Sistem telah diuji menggunakan 66 citra wajah dari 22 individu. Pengujian menggunakan citra wajah dalam keadaan normal menghasilkan rata-rata akurasi pengenalan sebesar 98.33% sedangkan pengujian menggunakan citra wajah dengan gangguan *noise* menghasilkan rata-rata akurasi pengenalan sebesar 86,66%. Berdasarkan hasil pengujian tersebut, maka dapat disimpulkan bahwa sistem pengenalan wajah yang dibangun menggunakan metode *Linear Discriminant Analysis* dan *k nearest neighbor* mampu melakukan pengenalan wajah dengan akurasi pengenalan yang baik, yaitu mencapai 98.33%.

V. SARAN

Adapun saran untuk penelitian tentang pengenalan wajah menggunakan metode *Linear Discriminant Analysis* dan *k nearest neighbor* yang selanjutnya

yaitu perlu dilakukan penelitian untuk perbandingan hasil klasifikasi dengan menggunakan metode klasifikasi yang lain, seperti *Fuzzy k-Nearest Neighbor* dan SVM (*Support Vector Machine*).

DAFTAR PUSTAKA

- [1] S.Z. Li, dan A. Jain, *Encyclopedia of biometrics*. Springer Publishing Company, Incorporated, 2015.
- [2] F. Mahmud, M. T. Khatun, S. T. Zuhori, S. Afroge, M. Aktar, dan B. Pal, "Face recognition using Principle Component Analysis and Linear Discriminant Analysis," *Electrical Engineering and Information Communication Technology (ICEEICT)*, 2015 International Conference on, pp. 1-4. IEEE, 2015.
- [3] P.S. Hiremath dan M. Hiremath, "Linear Discriminant Analysis for 3D Face Recognition Using Radon Transform," in Swamy P., Guru D. (eds) *Multimedia Processing, Communication and Computing Applications*, Lecture Notes in Electrical Engineering, vol 213. Springer, New Delhi, 2013.
- [4] M. Li, dan B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognition Letters* 26.5 (2005): 527-532.
- [5] H. Ebrahimpour, dan A. Kouzani, "Face Recognition Using Bagging KNN," *International Conference on Signal Processing and Communication Systems (ICSPCS'2007)* Australia, Gold Coast, 2007.
- [6] J.Y. Sari, *Pengenalan wajah pada citra digital menggunakan algoritma Eigenface dan Euclidean distance*, Skripsi, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari, 2012.
- [7] P.N. Belhumeur, J.P. Hespanha, dan D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1997, 19(7), 711-720.
- [8] R.A. Saragih, *Pengenalan Wajah Menggunakan Metode Fisherface*, *Jurnal Teknik Elektro*, 7(1), No.1, Maret 2007. Universitas Kristen Maranatha, Bandung.
- [9] N.A. Singh, M.B. Kumar, dan M.C. Bala, Face Recognition System based on SURF and LDA Technique. *International Journal of Intelligent Systems and Applications*, 8(2), 13, 2016.
- [10] M.C. Thomas dan E.H. Peter, Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 1967, IT-13:21-27.