

JURNAL INFORMATIKA Vol. 11, No. 2 Juli 2017

Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan K Nearest Neighbor

¹Fandiansyah, ²Jayanti Yusmah Sari, ³Ika Purwanti Ningrum

Jurusan Teknik Informatika, Fakultas Teknik

Universitas Halu Oleo, Kendari, 93232

Email: ¹lastfandiansyah@gmail.com,

²jayanti.yusmah.sari@gmail.com, ³ika.purwanti.n@gmail.com

Abstrak

Pengenalan wajah merupakan sistem biometrika yang banyak digunakan untuk pengenalan individu pada penggunaan mesin absensi atau akses kontrol. Hal ini karena wajah merupakan salah satu ciri biometrika yang paling umum digunakan untuk mengenali seseorang. Selain itu, pengenalan wajah tidak mengganggu kenyamanan seseorang saat pengambilan citra. Namun, ada dua hal yang menjadi masalah pengenalan wajah yaitu proses ekstraksi fitur dan teknik klasifikasi yang digunakan. Penelitian ini menggunakan linear discriminant analysis (LDA) dan k nearest neighbor untuk membangun sistem pengenalan wajah. LDA digunakan untuk membentuk satu set fisherface, di mana seluruh citra wajah direkonstruksi dari kombinasi fisherface dengan bobot yang berbeda-beda. Nilai bobot suatu citra testing akan dicocokkan dengan nilai bobot citra di database menggunakan metode klasifikasi k nearest neighbor. Sistem ini dibangun menggunakan bahasa pemrograman Java. Sistem telah diuji menggunakan database sebanyak 66 citra wajah dari 22 individu. Hasil pengujian menunjukkan metode LDA dan k nearest neighbor cukup optimal untuk melakukan pengenalan wajah dengan akurasi pengenalan citra wajah normal mencapai 98.33% dan akurasi pengenalan citra wajah yang diberi noise sebesar 86,66%.

Kata Kunci: *face recognition, k nearest neighbor, linear discriminant analysis.*

1. PENDAHULUAN

Sistem biometrika merupakan teknologi pengenalan diri menggunakan bagian tubuh manusia seperti sidik jari, telinga, wajah, geometri tangan, telapak tangan, retina, gigi dan bibir. Pengenalan wajah merupakan sistem biometrika yang banyak digunakan untuk identifikasi personal pada penggunaan mesin absensi atau akses *control*. Hal ini karena wajah merupakan salah satu biometrika yang paling umum digunakan untuk mengenali seseorang. Selain itu, pengenalan wajah juga tidak mengganggu kenyamanan seseorang saat pengambilan citra.

Banyak orang yang mencoba untuk membangun program aplikasi pengenalan wajah dengan berbagai macam metode yang masing-masing memiliki kelebihan dan kelemahan. Ada metode yang cepat dalam mengenali citra wajah akan tetapi mengorbankan tingkat keakuratan dalam pengenalan. Ada dua hal yang menjadi masalah utama pada pengenalan wajah yaitu proses ekstraksi fitur dari citra wajah dan juga teknik klasifikasi yang digunakan untuk mengklasifikasikan wajah yang ingin dikenali berdasarkan fitur-fitur yang telah dipilih.

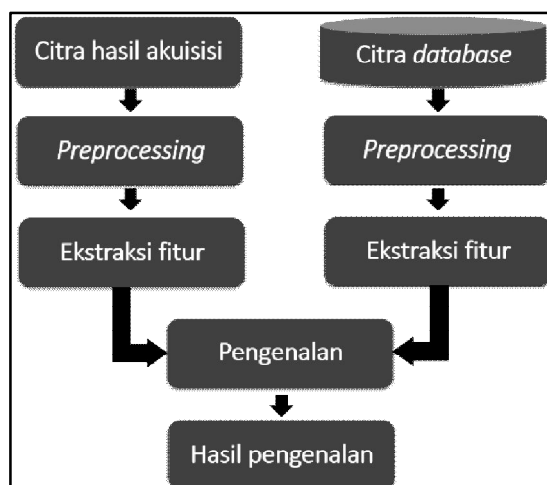
Ada beberapa metode ekstraksi fitur yang telah digunakan untuk aplikasi pengenalan wajah salah satunya yaitu *Principal Component Analysis* (PCA). Metode PCA bertujuan untuk memproyeksikan data pada arah yang memiliki variasi terbesar, yang ditunjukkan oleh vektor eigen yang bersesuaian dengan nilai terbesar dari matrikskovarian. Di samping itu juga bertujuan untuk mereduksi dimensi dengan melakukan transformasi linear dari suatu ruang berdimensi tinggi ke dalam ruang berdimensi rendah. Kelemahan dari metode PCA adalah kurang optimal dalam pemisahan antar kelas sehingga dapat mempengaruhi tingkat akurasi pengenalan wajah [1]. Untuk mengatasi masalah tersebut, Turk dan Pentland pada tahun 1991 [2] memperkenalkan *Linear Discriminant Analysis* (LDA) untuk pengenalan wajah. Metode ini mencoba menemukan subruang linear yang memaksimalkan perpindahan 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*.

Dalam pengenalan wajah, proses klasifikasi sama pentingnya dengan proses ekstraksi fitur. Setelah fitur-fitur penting citra wajah dihasilkan pada proses ekstraksi fitur, fitur-fitur tersebut nantinya akan digunakan untuk proses klasifikasi. Metode klasifikasi yang akan digunakan adalah klasifikasi *knearest neighbor*. Klasifikasi *knearest neighbor* adalah suatu metode yang menggunakan algoritma *supervised* di mana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada *knearest neighbor*. Klasifikasi *knearest neighbor* mempunyai kemampuan yang baik dalam mencari nilai kedekatan antara data baru dan data lama, yaitu berdasarkan pada pencocokkan bobot sejumlah fitur, meskipun data *training* kehilangan beberapa informasi yang diakibatkan oleh sistem yang tidak diketahui oleh pemakai [3]. Penelitian ini menggunakan *Linear Discriminant Analysis* (LDA) dan *knearest neighbor* untuk pengenalan wajah. LDA digunakan sebagai metode ekstraksi fitur untuk mengoptimalkan pemisahan antar kelas citra wajah sedangkan klasifikasi *knearest neighbor* mempunyai kemampuan yang baik dalam mencari nilai kedekatan antara data baru dan data lama, yaitu berdasarkan pada pencocokkan bobot sejumlah fitur.

2. METODE PENELITIAN

Pada penelitian ini dibuat sistem pengenalan wajah menggunakan metode *Linear Discriminant Analysis* (LDA) dan *knearest neighbor* yang 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 sebelumnya sudah tersimpan di dalam *database*.

Citra yang digunakan sebanyak 66 citra wajah dengan format **.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 pengenalan wajah yang dibuat.



Gambar 1. Gambaran umum sistem

Secara umum, tahapan pada sistem pengenalan wajah ini dibagi menjadi dua yaitu tahap *preprocessing*, dan 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*.

2.1 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.

- Akuisisi citra wajah menggunakan *webcam* laptop. Citra hasil akuisisi merupakan citra RGB 24 bit berformat PNG dengan ukuran 92 x 112 piksel.
- Konversi citra wajah hasil akuisisi dari RGB ke *grayscale*.
- 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.

2.2 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 *knearest neighbor*. Setelah melakukan tahapan *preprocessing*, maka selanjutnya adalah proses ekstraksi fitur citra seperti yang ditunjukkan pada Gambar 2. 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.

2.2.1 PCA (Principal Component Analysis) [4]

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 merupakan nilai piksel citra. Berikut uraian langkah-langkah Algoritma *Eigenface* seperti yang ditunjukkan pada Gambar 2 untuk memperoleh bobot dari citra *database*.

- a. Mengubah M jumlah citra di *database* dengan ukuran $B \times K$ (92×112) 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).

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

- 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 .

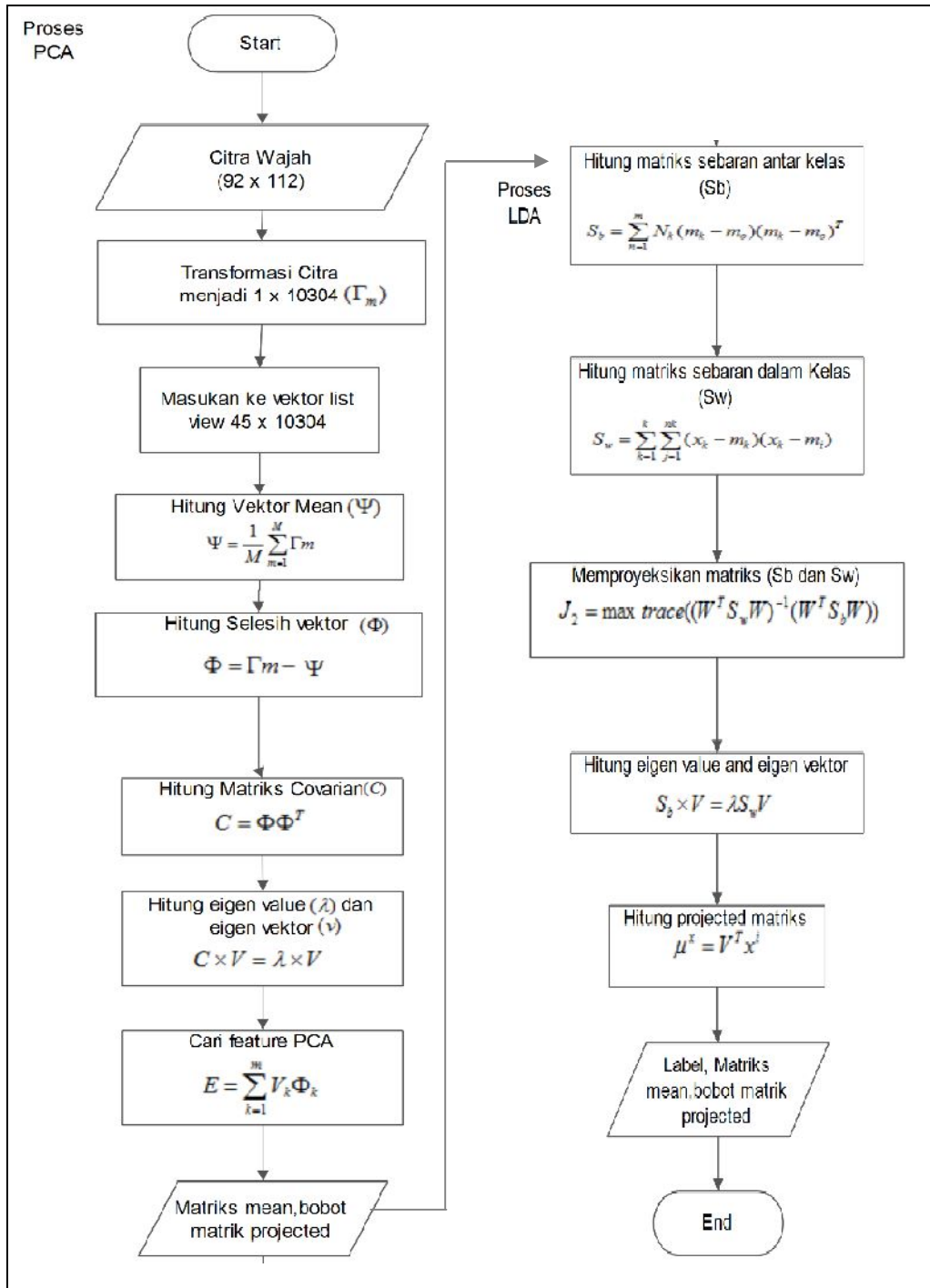
$$\Psi = \frac{1}{M} \sum_{m=1}^M P_m ; m = 1, 2, \dots, M \quad (4)$$

- 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 .

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



Gambar2. Proses pembentukan vektor fitur citra database

- 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 .

$$C V = \lambda V \quad (7)$$

$$(C - \lambda I)V = 0 \quad (8)$$

$$(\Phi \Phi^T - \lambda I)V = 0 \quad (9)$$

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 $M \times BK$.

$$E_k = \sum_{k=1}^M v_k \Phi_k ; k = 1, 2, \dots, M \quad (10)$$

- h. Normalisasi *eigenface* dengan menggunakan persamaan (11), dan simpan *eigenface* hasil normalisasi ini ke dalam matriks E_n berukuran 45×10304 pada *feature extraction*.

$$E_{nk} = \frac{E_k}{\sqrt{E_k \cdot E_k^T}} \quad (11)$$

- i. Hitung bobot dengan mengalikan matriks selisih Φ_k berukuran $M \times BK$ dengan matriks *transpose* dari E_{nk} yang berukuran $BK \times M$, menggunakan persamaan (12).

$$\omega = \Phi_k \times E_{nk}^T \quad (12)$$

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

2.2.2LDA (Linear Discriminant Analysis)

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 [5].

- a. Hasil dari bobot *eigenfaces* (E_k) dijadikan sebagai input yang akanditransformasikan 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).

$$S_b = \sum_{i=1}^k n_i(m_i - m_0)(m_i - m_0)^T \quad (13)$$

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

$$S_w = \sum_{i=1}^k \sum_{j=1}^{n_i} (x_i^{(j)} - m_i)(x_i^{(j)} - m_i)^T \quad (14)$$

- e. Memproyeksikan matriks sebaran dalam kelas. Matriks sebaran dalam kelas (S_w) adalah jarak matriks dalam kelas yang sama, menggunakan persamaan (15).

$$J_2(W) = \max_{\text{trace}} ((W^T S_w W)^{-1} (W^T S_b W)) \quad (15)$$

- f. Mencari *eigen value* (λ) dan nilai *eigenvector* (v) menggunakan persamaan (16).

$$S_b v = \lambda S_w v \quad (16)$$

- 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 *centerend image*) ke *fisher basis* vektor dengan menghitung *dot product* dari citra asal V^T ke tiap-tiap *fisher basis vector* x^i menggunakan persamaan (17).

$$\bar{u}^x = V^T x^i \quad (17)$$

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

2.3 Proses Pencocokan Fitur

Proses pencocokan fitur hasil ekstraksi menggunakan metode klasifikasi *knearestneighbor*. Pencocokan berguna untuk menentukan kelas dari suatu citra *testing* berdasarkan ciri-ciri yang telah diekstraksi. 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 [6][7]:

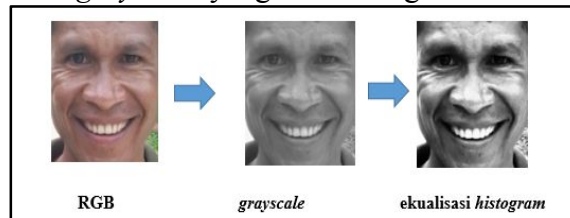
- a. Menentukan nilai k .
- b. Menghitung jarak antara citra *testing* dengan seluruh citra pada *database* menggunakan persamaan *euclidean distance*, persamaan (18) dan menentukan citra terdekat dengan citra *testing* berdasarkan nilai k .

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (18)$$

- c. Menentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak.
- d. Jika terjadi konflik atau keadaan seimbang pada kelas dengan jumlah anggota yang sama maka digunakan pemecahan konflik.

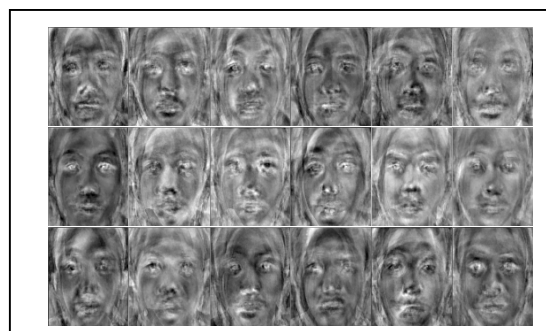
3. HASIL DAN PEMBAHASAN

Preprocessing dalam penelitian ini meliputi akuisisi citra, konversi citra RGB-*grayscale* dan ekualisasi *histogram*. Gambar 3 menunjukkan citra hasil dari tahap *preprocessing* berupa citra *grayscale* yang telah mengalami ekualisasi *histogram*.



Gambar 3. Contoh citra hasil *tahapan preprocessing*

Setelah melalui tahap *preprocessing*, 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.

Pengujian sistem pengenalan wajah menggunakan metode LDA dan k -NN pada penelitian ini dilakukan dengan dua skenario uji coba, dengan menggunakan 20 citra wajah dari individu yang sudah disimpan terlebih dahulu di dalam *database*. Pengujian pertama menggunakan citra uji normal, yaitu citra wajah hasil akuisisi dengan *pose* dan ekspresi berbeda-beda. Hasil dari pengujian ini ditunjukkan pada Tabel 1. 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. Adapun untuk mengetahui tingkat performansi dari metode LDA, dilakukan pula pengujian kedua dengan menggunakan citra wajah yang diberi *noise*. *Noise* dibuat dengan menggunakan bantuan Adobe Photoshop Cs4, dalam hal ini menggunakan *ocean distortion*. Hasil dari pengujian dapat dilihat pada Tabel 2, yaitu terdapat 8 kesalahan pengenalan saat uji coba kedua menggunakan citra wajah yang diberiefek 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 $k=7$. Pada Tabel 1 dan Tabel 2, hasil pengenalan wajah yang tidak tepat atau dikenali sebagai wajah dari individu lain, ditandai dengan kotak merah.

Tabel 1. Hasil Pengujian Menggunakan Citra Uji Normal

No	Citra Testing	Hasil pengenalan menggunakan k NN disertai jarak (d)			No	Citra Testing	Hasil pengenalan menggunakan k NN		
		$k=3$	$k=5$	$k=7$			$k=3$	$k=5$	$k=7$
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 Menggunakan Citra Uji Dengan Penambahan Gangguan *Ocean Distortion*

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

Tabel 3. Akurasi Pengenalan Wajah Menggunakan Citra Uji Normal

Nilai k	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. Akurasi Pengenalan Wajah Menggunakan Citra Uji Dengan Efek *Ocean Distortion*

Nilai k	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%

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 (19).

$$Akurasi = \frac{Total\ Pengenalan\ benar}{Total\ Citra\ Uji} \times 100\% \tag{19}$$

Hasil pengujian menggunakan citra wajah normal ditunjukkan pada Tabel 3. Adapun pengujian kedua yang dilakukan menggunakan citra uji yang telah diberi gangguan *ocean distortion* ditunjukkan pada Tabel 4. Pengujian kedua dilakukan untuk mengetahui performansi metode LDA dan k -NN dalam melakukan pengenalan citra wajah yang memiliki *noise*. Hasil pengujian pada Tabel 3 dan Tabel 4 menunjukkan metode LDA dan k nearest neighbor cukup optimal untuk melakukan pengenalan wajah dengan akurasi pengenalan citra wajah normal mencapai 100% dan akurasi pengenalan citra wajah yang diberi *noise* mencapai 90%.

Pada pengujian pertama dan kedua digunakan tiga nilai k yang berbeda yaitu $k=3, 5$, dan 7 untuk melihat pengaruh nilai k pada k -NN terhadap akurasi pengenalan wajah pada penelitian ini. 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 pada Tabel 4 dapat dilihat bahwa nilai akurasi tertinggi yaitu 90% untuk pengujian kedua diperoleh pada nilai $k=3$. Hasil kedua pengujian ini tidak berarti bahwa nilai $k=3$ lebih baik dibandingkan dengan $k=5$ atau $k=7$. Hal ini hanya menunjukkan bahwa nilai k yang optimal untuk 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%. Hal ini menunjukkan bahwa metode LDA cukup optimal dalam pengenalan citra wajah, namun masih memiliki keterbatasan dalam pengenalan citra wajah yang memiliki *noise*.

4. KESIMPULAN

Berdasarkan analisis hasil pengujian yang dilakukan maka dapat disimpulkan bahwa sistem pengenalan wajah menggunakan metode *linear discriminant analysis* dan *k nearest neighbor* dalam penelitian ini cukup optimal dalam melakukan pengenalan wajah dengan akurasi mencapai 98.33%.

5. SARAN

Adapun saran untuk penelitian selanjutnya terkait pengenalan wajah menggunakan metode *Linear Discriminant Analysis* dan *k nearest neighbor* yaitu perlu dilakukan perbandingan hasil klasifikasi dengan menggunakan metode klasifikasi yang lain, seperti *Fuzzy k-Nearest Neighbor* dan SVM (*Support Vector Machine*).

6. DAFTAR PUSTAKA

- [1] Belhumeur, P.N., Hespanha, J. P., dan Kriegman, D.J., 1997, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7), 711-720.
- [2] Turk, M., dan Pentland, A. 1991. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71-86.
- [3] Thomas M.C and Peter E.H., 1967, Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13:21-27.
- [4] Sari, J. Y., 2012, *Pengenalan wajah pada citra digital menggunakan algoritma Eigenface dan Euclidean distance*, Skripsi, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari.
- [5] Saragih, R. A., 2007, Pengenalan Wajah Menggunakan Metode Fisherface, *Jurnal Teknik Elektro*, 7(1), No.1, Maret 2007. Universitas Kristen Maranatha, Bandung.
- [6] Singh, S., et al., 2006, *Nearest Neighbour Strategies for Image Understanding*, www.dcs.ex.ac.uk/research/pann/pdf/pann_SS_004.pdf, Juli 2006.
- [7] Agusta, Y. 2007. "K-Means – Penerapan, Permasalahan dan Metode Terkait". <http://datamining.japati.net/dataupload/hadi1187726593.pdf>. Diakses pada 10 November 2015.