

## **Face Recognition Menggunakan Metode Direct GLCM dan K-NN**

I Komang Astina Adiputra<sup>1)</sup>, Raditiana Patmasari<sup>2)</sup>, Rita Magdalena<sup>3)</sup>

<sup>1),2),3)</sup>Teknik Telekomunikasi, Universitas Telkom  
Jl. Telekomunikasi Terusan Buah Batu Bandung 40257 Indonesia  
Email : komangastina77@gmail.com

**Abstrak.** Paper ini memperkenalkan metode face recognition baru yang didasarkan pada gray-level co-occurrence matrix (GLCM). Metode ini secara langsung menggunakan GLCM dengan mengubah matriks menjadi vektor yang dapat digunakan sebagai vektor fitur untuk proses klasifikasi, metode ini disebut direct GLCM. Proses klasifikasi yang digunakan yaitu K-Nearest Neighbor(K-NN), di mana dalam proses klasifikasi ini membandingkan antara fitur-fitur yang terdapat dalam K-NN yaitu Euclidean distance, Cityblock, Chebychev, dan Mincowski. Hasilnya menunjukkan bahwa menggunakan direct GLCM sebagai vektor fitur dalam proses pengenalan dengan menggunakan klasifikasi K-NN dengan fitur Cityblock menghasilkan akurasi 84,29%, FAR 6,67% dan FRR 9,05%.

**Kata kunci:** Face recognition , gray-level co-occurrence matrix, K-Nearest Neighbor, Cityblock.

### **1. Pendahuluan**

Face recognition adalah teknik *biometric* yang membuat suatu komputer atau mesin autentik lainnya dapat mengenal wajah seseorang. Prinsip dasar dari *face recognition* yaitu mengambil karakteristik alami dari wajah seseorang yang nantinya akan dibandingkan dengan karakteristik alami wajah yang ada pada basis data [1].

Face Recognition selalu menarik perhatian para peneliti sebagai salah satu teknik terpenting untuk identifikasi manusia. Salah satu keterbatasan sistem pengenalan secara *real-time* adalah kerumitan komputasional dari pendekatan yang ada. Banyak sistem dan algoritma telah diperkenalkan dalam beberapa dekade terakhir dengan *recognition rates* yang tinggi. Maka, pada penelitian ini metode ekstraksi citra wajah yang digunakan adalah *direct GLCM* karena mempunyai solusi dari permasalahan diatas yaitu metode ini sangat kompetitif dan melebihi teknik pengenalan wajah yang canggih seperti PCA dan LDA [2]. Menggunakan sejumlah kecil tingkat abu-abu membuat algoritma lebih cepat dan pada saat yang sama metode ini mempertahankan tingkat akurasi pengenalan yang tinggi[3].

*Direct GLCM* merupakan metode yang dikembangkan dari metode ekstraksi GLCM dengan menggunakan fitur hiralick yaitu *Energy, Entropy, Contrast, Variance, Homogeneity, Correlation, Sum Average, Sum Entropy, Sum Variance, Difference Variance, Difference Entropy, Maximum Correlation Coefficient, Information Measures of Correlation* [4]. Metode ini secara langsung menggunakan GLCM dengan mengubah matriks menjadi vektor yang dapat digunakan sebagai vektor fitur untuk proses klasifikasi [2][3].

Metode klasifikasi yang digunakan adalah *K-Nearest Neighbor* (K-NN), di mana dalam proses klasifikasi ini akan membandingkan antara fitur klasifikasi yang menyebabkan tingkat akurasi pengenalan wajah yang tinggi. Fitur yang digunakan dalam proses klasifikasi yaitu *Euclidean distance, Cityblock, Chebychev, dan Mincowski* [5].

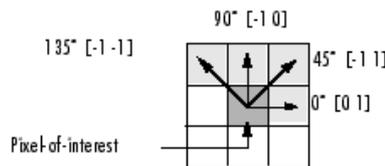
### **2. Pembahasan**

#### **2.1. Gray Level Co-occurrence Matrix**

Gray Level Co-occurrence Matrix (GLCM) adalah deskriptor tekstur yang banyak digunakan dan terbukti bahwa hasil yang diperoleh dari matriks *co-occurrence* lebih baik daripada metode diskriminasi tekstur lainnya. GLCM menghitung fitur statistik berdasarkan intensitas tingkat abu-abu citra. Fitur seperti GLCM berguna dalam pengenalan tekstur, segmentasi citra, pengambilan citra, analisis citra warna, klasifikasi citra, pengenalan objek dan metode analisis tekstur dll [6]. Salah satu pendekatan paling sederhana untuk mendeskripsikan tekstur adalah menggunakan momen statistik dari histogram intensitas suatu citra atau wilayah [7]. Menggunakan histogram hanya dalam perhitungan, akan menghasilkan ukuran tekstur yang hanya membawa informasi tentang distribusi intensitas, tetapi bukan tentang posisi relatif piksel terhadap satu sama lain dalam tekstur itu. Menggunakan pendekatan statistik seperti matriks *co-occurrence* akan membantu untuk memberikan informasi berharga tentang posisi relatif dari piksel tetangga dalam suatu citra [2]. Dengan diberikan citra I, ukuran  $N \times N$ , *co-occurrence* matriks P dapat didefinisikan sebagai berikut:

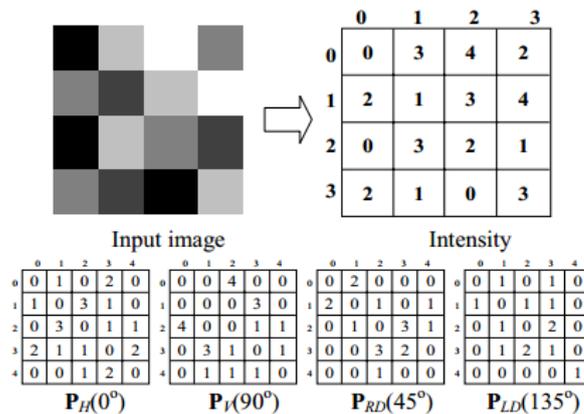
$$P(i, j) = \sum_{x=1}^N \sum_{y=1}^N \begin{cases} 1, & \text{if } I(x, y) = i, I(x + \Delta_x, y + \Delta_y) = j \\ 0, & \text{otherwise} \end{cases} \dots\dots\dots (1)$$

Dimana *offset*  $(\Delta_x, \Delta_y)$ , adalah jarak antara *pixel-of-interest* dan tetangganya, untuk ilustrasi *array* dari *offset* dapat dilihat pada Gambar 1. Perhatikan bahwa parameter *offset* membuat matrik *co-occurrence* sangat sensitif dengan rotasi [2].



Gambar 1. Ilustrasi *array*: *offset* = [ 0 1; -1 1; -1 0; -1 -1] [8].

Gambar 2 menunjukkan bagaimana menghasilkan empat matriks *co-occurrence* megunakan tingkat  $N_g = 5$  dan *offset* {[0 1], [-1 1], [-1 0], [-1 -1]} diartikan sebagai satu piksel yang berdekatan dengan empat arah yang ditentukan.



Gambar 2. Matriks *co-occurrence* dengan tingkat  $N_g=5$  dan empat *offsets*:  $P_H(0^\circ)$ ,  $P_V(90^\circ)$ ,  $P_{RD}(45^\circ)$ , dan  $P_{LD}(135^\circ)$  [2]

Rata-rata matriks *gray-level co-occurrence* dapat dihitung dengan:

$$P = \frac{P_H + P_V + P_{RD} + P_{LD}}{4} \dots\dots\dots (3)$$

**2.2. K-Nearest Neighbor**

*K-Nearest Neighbor* atau K-NN adalah algoritma *instance based learning* atau *case-based reasoning*. Dimana *case-based reasoning* sendiri adalah sebuah pendekatan penyelesaian masalah dengan cara memanfaatkan kondisi yang telah dilakukan sebelumnya. KNN bisa dikatakan sebagai algoritma *supervised learning* dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari

kategori K tetangga terdekat. K-NN digunakan dalam banyak aplikasi data *mining*, *statistical pattern recognition*, *image processing*, dan lain sebagainya. Tujuan dari algoritma ini adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dan sampel-sample dari data *training* [9].

*K-Nearest Neighbor* adalah salah satu metode klasifikasi yang paling sederhana. Di sisi lain, K-NN adalah algoritma pembelajaran semi-supervisi yang memerlukan data pelatihan dan nilai K yang telah ditentukan untuk mengetahui data K terdekat berdasarkan perhitungan jarak [10].

Dalam pengklasifikasiannya, algoritma K-NN menggunakan *neighborhood classification* sebagai nilai prediksi dari nilai *instance* yang baru. K-NN bekerja berdasarkan jarak minimum dari data baru ke sample data latih untuk menentukan K tetangga terdekat. Setelahnya akan kita dapatkan nilai mayoritas sebagai hasil prediksi dari data yang baru tersebut.

Langkah – langkah dari algoritma K – Nearest Neighbor (KNN) [9]:

- Tentukan parameter K = jumlah banyaknya tetangga terdekat.
- Hitung jarak antar data baru dan semua data yang ada di data latih.
- Urutkan jarak tersebut dan tentukan tetangga mana yang terdekat berdasarkan jarak minimum ke – K.
- Tentukan kategori dari tetangga terdekat.
- Gunakan kategori mayoritas yang sederhana dari tetangga yang terdekat tersebut sebagai nilai prediksi dari data yang baru.

Perhitungan jarak pada metode K-NN dapat dilakukan menggunakan rumus Euclidean distance, Cityblock, Chebychev, dan Mincowski [5].

- *Euclidean Distance*:

$$j(v_1, v_2) = \sqrt{\sum_{k=1}^N (v_1(k) - v_2(k))^2} \dots\dots\dots(4)$$

- *Cityblock* atau *manhattan distance*

$$j(v_1, v_2) = \sum_{k=1}^N |v_1(k) - v_2(k)| \dots\dots\dots(5)$$

- *Chebychev*

$$j(v_1, v_2) = \max_{k=1 \rightarrow N} (|v_1(k) - v_2(k)|) \dots\dots\dots(6)$$

- *Minkowski*

$$j(v_1, v_2) = \sqrt[p]{\sum_{k=1}^N (v_1(k) - v_2(k))^p} \dots\dots\dots(7)$$

dengan:

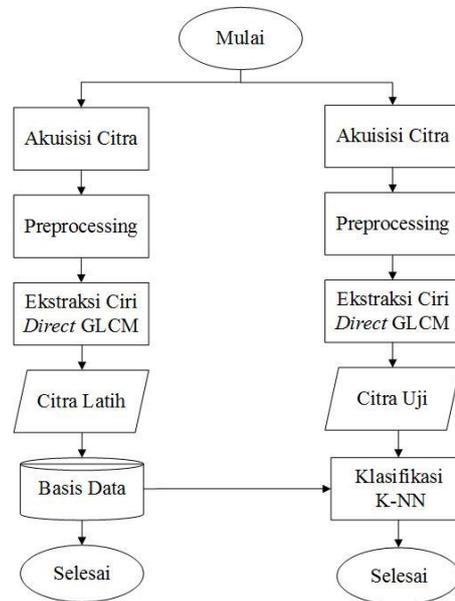
$j$  : jarak data uji ke data latih

$v_1(k)$ : fitur data uji ke-k, dengan  $k = 1, 2, 3, \dots, N$

$v_2(k)$ : fitur data latih ke-k, dengan  $k = 1, 2, 3, \dots, N$

### 2.3. Perancangan Sistem

Diagram alir sistem terdapat dua proses yaitu proses pelatihan data yang disimpan pada basis data dan proses pengujian data yang diakhiri dengan proses klasifikasi. Diagram alir sistem dapat dilihat pada Gambar 3.

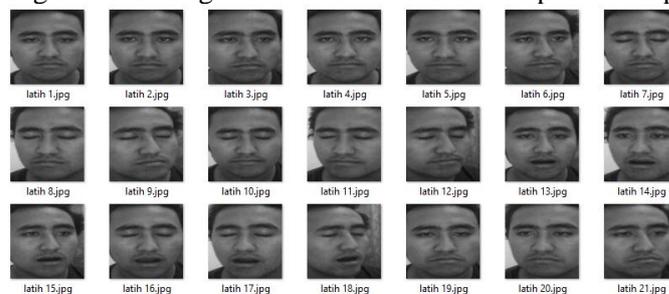


Gambar 3. Diagram alir sistem proses latih dan proses uji

Diagram alir sistem pada proses pelatihan data dimulai dari akuisisi citra wajah dengan menggunakan kamera webcam, di mana data latih yang diperoleh dapat dilihat pada Gambar 4. Setelah melakukan proses akuisisi citra, maka dilanjutkan dengan *preprocessing*. *Preprocessing* yang dilakukan yaitu *crop* citra wajah, *resize*, dan *grayscale*. Proses berikutnya yaitu ekstraksi ciri, di mana metode ekstraksi ciri yang digunakan untuk citra wajah yaitu *direct GLCM*. Setelah ciri dari citra wajah diperoleh selanjutnya akan disimpan di basis data. Untuk proses pengujian mempunyai alur yang sama dengan proses pelatihan. Namun, pada proses pengujian ini terdapat tahapan mengklasifikasikan citra yang telah diambil cirinya dengan semua ciri yang ada pada basis data. Metode yang digunakan untuk mengklasifikasikan citra adalah K-NN, dengan membandingkan fitur terbaik untuk menghasilkan akurasi *face recognition* yang bagus. Adapun fitur yang digunakan yaitu lain *Euclidean*, *Cityblock*, *Chebychev*, dan *Minkowski*.

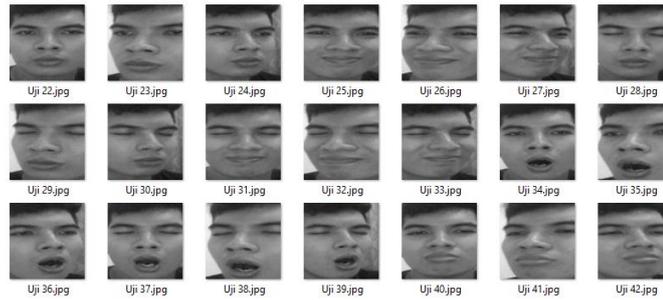
## 2.4. Hasil Simulasi

Simulasi telah dilakukan dengan data set yang telah diambil dalam proses akuisisi citra dengan ukuran  $92 \times 90$ . Jumlah citra yang digunakan sebagai data latih adalah 168 citra dimana terdiri 8 kelas yang berbeda atau 8 orang yang berbeda dengan 21 kondisi. Data latih dapat dilihat pada Gambar 4.



Gambar 4. Sampel citra latih

Adapun jumlah data uji yang digunakan adalah 210 citra dalam 10 kelas yang berbeda dengan 21 kondisi. Dari 10 kelas citra uji, 8 kelas (168 citra) terdapat pada citra latih dan 2 kelas (42 citra) diluar data latih. Untuk data uji dapat dilihat pada Gambar 5.

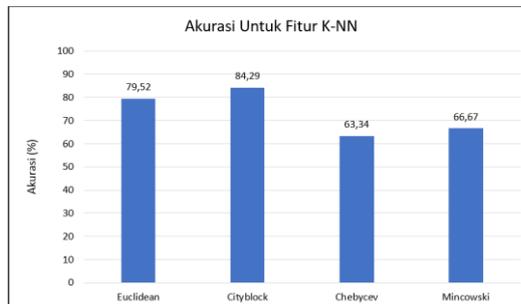


Gambar 5. Sampel citra uji

Pada tahapan *preprocessing* dilakukan proses *grayscale* yaitu proses merubah suatu citra RGB menjadi citra abu atau *grey*. Selai itu adapun proses mengubah ukuran citra atau *resize* dengan ukuran  $92 \times 90$ , tujuan dari *resize* yaitu menyeragamkan semua citra agar nantinya dalam ekstraksi tidak terjadi kesalahan atau *error*. Setelah melalui semua proses sesuai dengan diagram alir sistem, adapun keluaran sistem yaitu akurasi, FAR, dan FRR dari empat fitur K-NN yang akan dibandingkan. Ketetangaan terdekat pada proses klasifikasi yang digunakan yaitu 1 (K=1). Adapun perbandingan akurasi sistem dari keempat fitur K-NN dapat dilihat pada Tabel 1.

Tabel 1. Analisis Komperatif

Tingkat pengukuran	Fitur K-NN			
	<i>Euclidean</i>	<i>Cityblock</i>	<i>Chebychev</i>	<i>Minkowski</i>
Persentase dari tingkat pengukuran	79.52%	84.29%	63.34%	66.67%

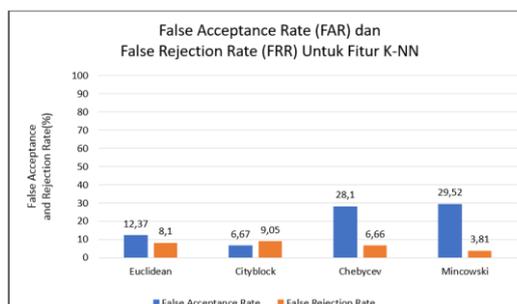


Gambar 6. Tingkat Pengukuran Untuk *Various* Fitur K-NN

Pada Gambar 6 menjelaskan nilai akurasi dari hasil ekstraksi menggunakan GLCM dan empat fitur K-NN yang digunakan yaitu *Euclidean* sebesar 79,52%, *Cityblock* 84,29%, *Chebychev* 63,34%, dan *Mincowski* 66,67%.

Tabel 2. Analisis Komperatif FAR & FRR

Tingkat pengukuran	Fitur K-NN			
	<i>Euclidean</i>	<i>Cityblock</i>	<i>Chebychev</i>	<i>Mincowski</i>
<i>False Acceptance Rate</i>	12,38%	6,67%	28,1%	29,52%
<i>False Rejection Rate</i>	8,1%	9,05%	6,67%	3,81%



Gambar 7. Tingkat Pengukuran Untuk FAR dan FRR  
Various Fitur K-NN

Pada Tabel 2 Dapat kita lihat perbandingan antara *False Acceptance Rate* dan *False Rejection Rete* dari empat fitur K-NN. Pada Gambar 7 menjelaskan nilai FAR dan FRR dari empat fitur K-NN yang digunakan yaitu *Euclidean* (FAR=12,38% & FRR=8,1%), *Cityblock* (FAR=6,67% & FRR=9,05%), *Chebyshev* (FAR=28,1% & FRR=6,67%), dan *Mincowski* (FAR=29,52% & FRR=3,81%). FAR menunjukkan persentase indentifikasi di mana orang yang tidak sah diterima secara tidak benar. FRR menunjukkan persentase indentifikasi di mana orang yang dikenal ditolak secara tidak benar.

### 3. Kesimpulan

Setelah perancangan dan implementasi terlaksana serta telah dilakukan pengujian *face recognition* maka dapat ditarik kesimpulan, dimana dengan menggunakan metode ekstraksi ciri *direct GLCM* dan menggunakan metode klasifikasi empat fitur K-NN (*Euclidean distance*, *Cityblock*, *Chebyshev*, dan *Mincowski*) mendapatkan akurasi yang terbaik dengan fitur *Cityblock*. Dengan menggunakan fitur *Cityblock* dalam tahapan klasifikasi menghasilkan akurasi terbaik yaitu 84,29% dengan FAR sebesar 6,67% dan FRR sebesar 9,05%. Maka pada proses *face recognition* menggunakan ekstraksi ciri *direct GLCM* dengan menggunakan 210 citra uji sesuai pada Gambar 5, direkomendasikan menggunakan metode klasifikasi K-NN dengan fitur *Cityblock*.

### Ucapan Terima Kasih

Terimakasih kepada Laboratorium Image Processing and Vision, Fakultas Teknik Elektro, Universitas Telkom, telah menyediakan fasilitas selama penelitian ini sehingga dapat terselesaikan.

### Daftar Pustaka

- [1]. O. N. Akbar, "Multi Face Recognition Menggunakan Complete Fuzzy Fisher Linear Discriminant," Telkom University, Bandung, 2012.
- [2]. E. Alaa, D. Hasan, "Co-Occurrence based Statistical Approach for Face Recognition", IEEE, Turkish Republic of Northern Cyprus, 2009.
- [3]. E. Alaa, D. Hasan, "Co-occurrence matrix and its statistical features as a new approach for face recognition," Tubitak, vol. 19, no. 1, p. 97 – 107, 2011.
- [4]. D. Gadkari, "Image Quality Analysis Using GLCM," University of Central Florida, Orlando, 2004.
- [5]. A. Kadir and A.Susanto, Teori dan Aplikasi Pengolahan citra. Yogyakarta: CV.Andi Offset, 2013.
- [6]. C. Nageswara Rao, S. Sreehari Sastry, K. Mallika, H. S. Tiong and K. Mahalaksmi. "Co-Occurrence Matrix and Its Statistical Features as an Approach for Identification Of Phase Transitions Of Mesogens," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 2, no. 9, p. 4531 – 4538, 2013.
- [7]. R. C. Gonzalez, & R. E. Woods, *Digital Image Processing*, 3rd Ed. Prentice Hall, 2008.
- [8]. "Graycomatrix" [Online]. Available: <https://www.mathworks.com/help/images/ref/graycomatrix.html>. [Accessed: 16-Sep-2018].
- [9]. S. N. Wibowo, "Identifikasi Jenis Batuan Beku Melihat Tekstur batuan Menggunakan Metode Discrete Wavalet Transform (DWT) Dan K-Nearest Neighbor," Telkom University, Bandung, 2017.
- [10]. S. S. Aung, I. Nagayama and S. Tamaki, "Regional Distance-based k-NN Classification," IEEE, Okinawa, 2017.