

Implementasi Algoritma *K-Nearest Neighbor* (K-NN) dalam Mengklasifikasikan Berbagai Jenis Ekspresi Wajah Manusia

Marinatul Khaliqah¹, Luluk Sarifah^{2*}, Siti Khotijah³

^{1,2,3} *Prodi Matematika, Fakultas MIPA, Universitas Annugayah, Indonesia*

*Penulis Korespondensi. Email: lu2ksarifah@istannugayah.ac.id

ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasikan ekspresi wajah manusia menggunakan algoritma *K-Nearest Neighbor* dengan dibantu ekstraksi ciri GLCM sebagai variabel yang dijadikan inputan pada proses klasifikasi. Ekspresi wajah manusia adalah bentuk respon alami yang ditunjukkan oleh manusia untuk memperlihatkan perasaan yang sedang dirasakan saat berinteraksi sosial. Pentingnya penelitian terkait ekspresi wajah adalah untuk membantu perkembangan teknologi di dunia dalam memainkan peran penting pada interaksi komputer dan manusia. Pada penelitian ini dilakukan dua jenis pengujian, yaitu dengan jumlah data yang lebih banyak dan jumlah data yang lebih sedikit dengan penggunaan $k = 1, k = 3$, dan $k = 5$ dengan tujuan mengetahui tingkat akurasi yang lebih baik dari algoritma *K-Nearest Neighbor*. Setelah dilakukan pengujian didapatkan hasil akurasi yang lebih baik dari algoritma *K-Nearest Neighbor* adalah dengan jumlah data yang lebih banyak dengan penggunaan $k = 1$, yaitu dengan tingkat akurasi mencapai 100%. Jadi dapat disimpulkan bahwa algoritma *K-Nearest Neighbor* dapat mencapai akurasi terbaiknya dalam mengklasifikasikan jenis ekspresi wajah manusia adalah dengan pengujian data yang lebih banyak dan dengan penggunaan $k = 1$.

Kata Kunci: Ekspresi Wajah; GLCM; Klasifikasi; *K-Nearest Neighbor*

ABSTRACT

This research aims to classify human facial expressions using the K-Nearest Neighbor algorithm assisted by GLCM feature extraction as variables used as input in the classification process. Human facial expressions are a form of natural response shown by humans to show the feelings they are feeling during social interaction. The importance of research related to facial expressions is to help the development of technology in the world to play an important role in computer and human interactions. In this research, two types of testing were carried out, namely with a larger amount of data and a smaller amount of data using $k = 1, k = 3$, and $k = 5$ with the aim of knowing the better level of accuracy of the K-Nearest Neighbor algorithm. After testing, we obtained better accuracy results from the K-Nearest Neighbor algorithm, namely with a larger amount of data using $k = 1$, namely with an accuracy level of up to 100%. So it can be concluded that the K-Nearest Neighbor algorithm can achieve the best accuracy in classifying types of human facial expressions by testing more data and using $k = 1$.

Keyword: Facial Expression; GLCM; Classification; *K-Nearest Neighbor*

Article info:

Submitted: 5 February 2024

Accepted: 24 May 2024

How to cite this article:

Khaliqah, M., Sarifah, L., & Khotijah, S. (2024). Implementasi Algoritma K-Nearest Neighbor (K-NN) dalam Mengklasifikasikan Berbagai Jenis Ekspresi Wajah Manusia. Zeta - Math Journal, 9(1), 10-20. <https://doi.org/10.31102/zeta.2024.9.1.10-20>



This article is an open access article distributed under the terms and conditions of the
[Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

1. PENDAHULUAN

Di dalam kehidupan sehari-hari khususnya dalam komunikasi yang berkaitan dengan hubungan antarpribadi, wajah sering digunakan dalam berekspresi. Melalui ekspresi wajah, maka akan terlihat emosi atau perasaan yang sedang dirasakan oleh individu (Ali, 2016). Ekspresi wajah merupakan salah satu bentuk komunikasi *non verbal* yang dapat menyampaikan keadaan emosional seseorang, sehingga ekspresi wajah merupakan cara penting untuk menyampaikan pesan sosial dalam kehidupan manusia. (Ahmad & Alhamad, 2019). Paul Ekman, seorang psikolog Amerika mendefinisikan enam kategori klasifikasi emosi, yaitu bahagia, sedih, terkejut, marah, takut, dan jijik. Sebagian besar sistem pengenalan ekspresi wajah mengklasifikasikan emosi ke dalam enam kategori universal tersebut (Abidin, 2012).

Penelitian tentang klasifikasi ekspresi wajah semakin marak dilakukan seiring perkembangan teknologi. Pengklasifikasian tersebut dilakukan melalui kecerdasan mesin atau komputer. Pentingnya penelitian terkait ekspresi wajah adalah membantu perkembangan teknologi di dunia dalam memainkan peran penting dalam interaksi komputer dan manusia. Pengenalan ekspresi wajah manusia ini bisa diaplikasikan pada perancangan robot, kamera pengawas keamanan, pengenalan *password* pada android, *game* interaktif, serta dapat digunakan untuk memainkan pemutar musik berdasarkan ekspresi wajah (Reynaldo & Maliki, 2021).

Klasifikasi merupakan proses pengelompokan fitur citra latih yang telah disimpan dengan fitur citra uji (Asmara et al., 2018). Klasifikasi ekspresi wajah manusia dapat dilakukan dengan menggunakan algoritma-algoritma yang telah teruji. Pada penelitian ini, algoritma yang akan diaplikasikan adalah algoritma *K-Nearest Neighbor* (K-NN). Algoritma tersebut akan digunakan untuk melihat nilai akurasi terbaik yang dapat dihasilkan dalam melakukan klasifikasi ekspresi wajah manusia.

Penelitian terkait ekspresi wajah dengan menggunakan algoritma *K-Nearest Neighbor* juga pernah dilakukan oleh peneliti sebelumnya. Seperti penelitian yang dilakukan oleh Ilmego, M. A. Z. (2020) dengan judul Implementasi Metode *K-Nearest Neighborhood* untuk Deteksi Emosi Pada Wajah. Hasil penelitian tersebut menunjukkan bahwa metode K-NN dapat melakukan klasifikasi dengan akurasi terbaik menggunakan penghitungan jarak *Euclidean*, dengan rata-rata tertinggi yaitu 0.8107 (81,07%). Sedangkan *Squared Chi-Square* dan *Manhattan* berturut-turut adalah 0.8067 (80,67%) dan 0,5697 (56,97%). Keterbaruan penelitian ini dari peneliti sebelumnya adalah penggunaan ekspresi yang diklasifikasikan ada 6 jenis ekspresi, yaitu ekspresi bahagia, sedih, marah, takut, terkejut, dan jijik. Sedangkan pada penelitian sebelumnya hanya menggunakan 2 ekspresi yaitu senang dan sedih. Selain itu, penulis juga menggunakan ekstraksi ciri GLCM dalam proses klasifikasi.

Berdasarkan penelitian tersebut, maka pada penelitian ini akan diuji terkait klasifikasi ekspresi wajah menggunakan metode K-NN, tujuannya untuk melihat penerapan dan kinerja dari algoritma tersebut dalam mengklasifikasikan ekspresi wajah manusia. Selain itu juga untuk melihat perbandingan penggunaan data yang lebih banyak dan lebih sedikit dari algoritma K-NN dalam mengklasifikasikan ekspresi wajah manusia.

2. TINJAUAN PUSTAKA

2.1 Ekspresi Wajah

Wajah merupakan salah satu alat atau instrumen yang dapat digunakan untuk mengenali seseorang. Dari wajah, manusia dapat menunjukkan ekspresi atau emosi dalam berinteraksi sosial (Seandrio et al., 2021). Ekspresi wajah merupakan perubahan bentuk raut muka seseorang dalam menanggapi keadaan perasaan dan komunikasi sosial. Melalui ekspresi wajah, secara tidak langsung memberi ungkapan akan suatu bentuk perasaan, keinginan, serta tujuan tertentu (Ihsan et al., 2021). Ekspresi wajah dapat dimiliki oleh manusia baik secara sengaja maupun tidak sengaja. Namun, pada umumnya ekspresi wajah dialami secara tidak sengaja karena adanya perasaan atau emosi yang sedang dirasakan oleh manusia. Dalam sebuah penelitian ditemukan bahwa ekspresi wajah dapat menyampaikan informasi hingga 55% dalam interaksi dibandingkan dengan *vocal* dan verbal secara keseluruhan yang hanya dapat memberikan informasi hingga 45% (Ahmad & Alhamad, 2019).

2.2 Citra Digital

Citra didefinisikan sebagai fungsi $f(x, y)$, berukuran M baris dan N kolom, dimana x dan y adalah koordinat spasial dari citra, sedangkan $f(x, y)$ merupakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Jika nilai x , y , dan f secara keseluruhan memiliki nilai yang berhingga dan bernilai diskrit, maka citra tersebut disebut citra digital (Susanto, 2019). Citra digital dapat diartikan sebagai hasil penangkapan suatu objek yang mempunyai nilai koordinat spasial dan juga memiliki intensitas kecerahan warna pada suatu objek. Citra merupakan kumpulan dari beberapa piksel yang disusun ke dalam *array* (larik) dua dimensi yang berisikan nilai-nilai sesungguhnya dan dipresentasikan ke dalam deretan bit tertentu (Zana et al., 2021).

2.3 Gray Level Co-Occurrence Matrix (GLCM)

Metode GLCM menurut Xie et al. (2010) merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi (Ahmad & Alhamad, 2019). Metode GLCM juga diartikan sebagai teknik untuk memperoleh statistik orde kedua dengan menghitung probabilitas hubungan kedekatan antara dua piksel dalam orientasi jarak spesifik dan derajat tertentu. Sudut yang digunakan adalah 0, 45, 90, 135 dan jarak yang digunakan tergantung dengan ketetapan antara piksel. Metode GLCM memiliki beberapa nilai fitur yang dihasilkan, seperti *Contrast*, *Correlation*, *Energy*, dan *Homogeneity* (Manuel & Ernawati, 2020). Metode *Gray Level Co-Occurrence Matrix* (GLCM) memiliki beberapa nilai fitur yang dihasilkan, yaitu :

a. *Contrast*

Merupakan salah satu variabel yang terdapat pada fitur GLCM dan berfungsi untuk mengukur intensitas kontras keabuan pada suatu citra. Untuk menghitung *Contrast* dapat dilakukan menggunakan persamaan berikut :

$$Con = \sum_i \sum_j (i - j)^2 p_{ij} \quad (1)$$

b. *Correlation*

Merupakan variabel yang terdapat pada fitur GLCM dan digunakan untuk Mengukur tingkat keterkaitan linear dari derajat dengan derajat keabuan citra. Untuk menghitung *Correlation* dapat dilakukan menggunakan persamaan berikut :

$$Cor = \sum_i \sum_j \frac{(i - \mu_i)(j - \mu_j)p(i,j)}{\sigma_i \sigma_j} \quad (2)$$

c. *Energy*

Merupakan jumlah dari nilai kuadrat pada GLCM yang telah melewati proses normalisasi. Untuk menghitung *Energy* dapat dilakukan menggunakan persamaan berikut :

$$Eng = \sum_i \sum_j p(i,j)^2 \quad (3)$$

d. *Homogeneity*

Merupakan nilai yang berhubungan dengan pengukuran kedekatan antara distribusi nilai elemen yang ada pada GLCM dengan diagonal GLCM. Untuk menghitung *Homogeneity* dapat dilakukan menggunakan persamaan berikut :

$$Hom = \sum_i \sum_j \frac{p(i,j)}{1 + |i - j|} \quad (4)$$

2.4 Klasifikasi

Klasifikasi adalah mempelajari struktur kumpulan data contoh, kemudian dipartisi menjadi kelompok-kelompok yang disebut sebagai kategori atau kelas (Mahran et al., 2020). Klasifikasi merupakan salah satu *task* analisis data dimana akan dibentuk sebuah model (*classifier*) untuk menyatakan suatu objek ke salah satu kategori atau kelas yang sudah didefinisikan sebelumnya. Klasifikasi data merupakan proses yang terdiri atas dua langkah. Langkah pertama adalah tahap pelatihan, dimana sebuah algoritma klasifikasi akan membentuk *classifier* dengan menganalisis atau belajar dari sekumpulan data latih. Sedangkan langkah kedua, model yang sudah terbentuk digunakan untuk mengklasifikasikan data uji. Akurasi dari sebuah *classifier* untuk sekumpulan data uji yang diberikan merupakan persentase dari data-data uji yang diklasifikasikan dengan benar oleh *classifier* (Asmara et al., 2018).

Klasifikasi juga diartikan sebagai suatu proses pembelajaran yang berfungsi untuk menentukan setiap himpunan atribut dari suatu objek atau dapat juga diartikan sebagai proses pengelompokan suatu data. Klasifikasi digunakan dalam menggambarkan sebuah dataset dimana setiap tipe data adalah berbentuk nominal atau biner. Pada klasifikasi data *supervised* akan dibagi menjadi dua, yaitu data *training* dan data *testing*, dimana data *training* akan dianalisis dengan menggunakan algoritma klasifikasi. Berikut beberapa algoritma klasifikasi yaitu algoritma *K-Nearest Neighbor* (K-NN), algoritma *Naïve Bayes*, algoritma C4.5, algoritma C5, dan sebagainya (Putry, 2022).

2.5 *K-Nearest Neighbor* (K-NN)

Algoritma *K-Nearest Neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (Nendya et al., 2021). Algoritma K-NN akan mengklasifikasikan objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. K-NN termasuk dalam golongan *supervised*, dimana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang sudah ada. Beberapa keunggulan dari algoritma ini adalah tangguh terhadap data *training* yang memiliki banyak *noise* dan keefektifan apabila data *training* besar. Selain itu, proses klasifikasi mudah untuk direpresentasikan dibandingkan dengan metode klasifikasi lain (Kusumawati, 2020).

Dalam Algoritma K-NN, kedekatan ketetanggaan akan dilihat melalui penghitungan jarak. Penghitungan jarak yang digunakan adalah *Euclidean Distance*. Jarak *Euclidean* dapat dicari dengan menggunakan persamaan berikut :

$$D_{xy} = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (5)$$

dimana :

D_{xy} : Jarak *Euclidean*

X : Data *training*

Y : Data *testing*

n : Jumlah atribut individu antara 1 sampai dengan n

i : Atribut individu antara 1 sampai dengan n (Syahid, 2021).

2.6 Confusion Matrix

Confusion Matrix adalah matriks yang berisi informasi tentang hasil prediksi klasifikasi dan data aktual yang dilakukan oleh sistem klasifikasi. Kinerja sistem klasifikasi umumnya dihitung menggunakan data dalam matriks ini (Andono, dkk., 2017). *Confusion Matrix* memiliki 4 pengujian yang berguna untuk mengukur performa sebuah model, yaitu *Accuracy*, *Precision*, *Recall*, dan *f1-Score*.

Accuracy dalam *Confusion Matrix* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Untuk menguji akurasi dalam *Confusion Matrix* menggunakan rumus berikut :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Precision dalam *Confusion Matrix* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Rumus *Precision* adalah sebagai berikut :

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall dalam *Confusion Matrix* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Rumus dari *Recall* adalah sebagai berikut :

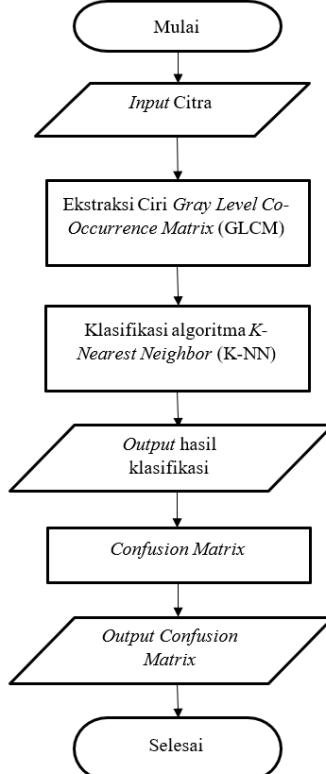
$$Recall = \frac{TP}{TP+FN} \quad (8)$$

f1-Score dalam *Confusion Matrix* merupakan rata-rata harmonik dari *Precision* dan *Recall*. Rumus dari *f1-Score* adalah sebagai berikut :

$$f1-Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

3. METODE PENELITIAN

Berikut diagram alir dari implementasi algoritma *K-Nearest Neighbor* (K-NN) dalam mengklasifikasikan berbagai jenis ekspresi wajah manusia yang dapat di lihat pada gambar 3.1



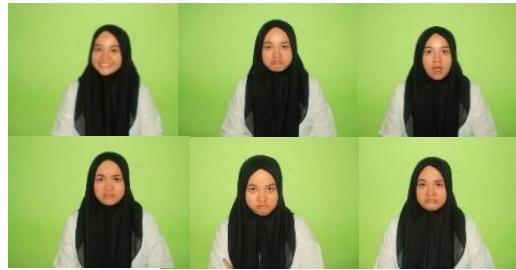
Gambar 3.1: Diagram alir klasifikasi ekspresi wajah dengan K-NN

Berdasarkan Gambar 3.1 dijelaskan mengenai alur analisis data pada proses klasifikasi dengan algoritma KNN, dimana proses tersebut dimulai dari memasukkan citra, baik citra pelatihan dan pengujian karena implementasi algoritma KNN pada klasifikasi ekspresi wajah terdiri dari dua tahapan yaitu *training* (pelatihan) dan *testing* (pengujian), selanjutnya melakukan ekstraksi ciri dengan metode GLCM, melakukan klasifikasi dengan metode K-NN, kemudian hasil dari klasifikasi diuji menggunakan *Confusion Matrix* untuk melihat tingkat akurasi dari algoritma klasifikasi yang diuji.

4. HASIL DAN PEMBAHASAN

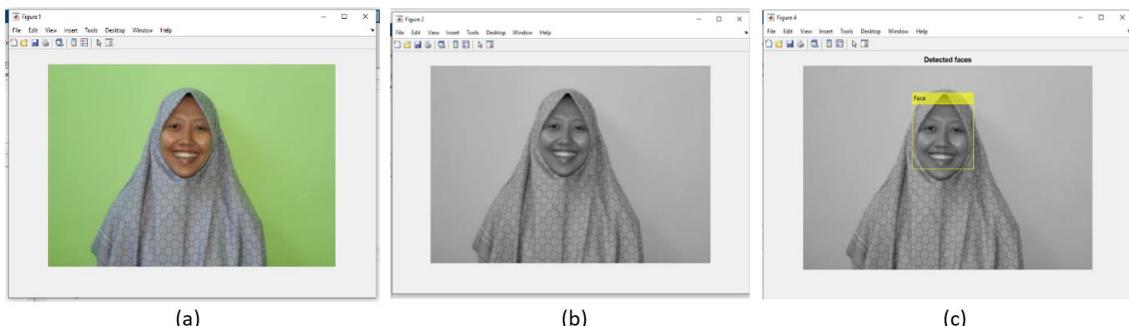
4.1 Data Simulasi

Data pada penelitian ini diambil dari ekspresi wajah mahasiswi Fakultas MIPA dan Fakultas Teknik IST Annuqayah angkatan 2019-2022. Pengumpulan data diambil menggunakan Camera DSLR, jarak 1 meter dengan latar yang sama yaitu warna hijau, dan pencahayaan berupa lampu kilat (blitz).



Gambar 4.1 Contoh pengambilan ekspresi wajah

Data ekspresi wajah tersebut kemudian diolah menggunakan Matlab dengan membaca citra RGB, mengubah citra RGB menjadi Grayscale, dan mendeteksi area wajah (*Detection Faces*) yang dapat di lihat pada gambar 4.2.



Gambar 4.2: (a). Citra RGB, (b). Citra Grayscale, (c) *Detection Face*

4.2 Analisa Hasil

Pada analisa hasil akan diperlihatkan kinerja metode *K-Nearest Neighbor* (K-NN). Kinerja tersebut dapat dilihat dari hasil akurasi algoritma K-NN dengan penggunaan $k = 1$, $k = 3$, dan $k = 5$. Selain itu juga akan dilihat hasil akurasi K-NN dengan penggunaan jumlah file pelatihan sebanyak 390 dan 138 dan penggunaan jumlah file pengujian sebanyak 102 dan 30.

4.2.1 Hasil Akurasi Algoritma K-NN

Implementasi algoritma K-NN pada klasifikasi ekspresi wajah terdiri dari dua tahapan yaitu sebagai berikut :

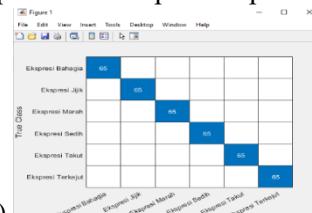
1. Pelatihan

a. Pelatihan algoritma K-NN dengan penggunaan $k = 1$ pada 390 data citra

Berikut hasil akurasi pelatihan yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 1$ pada 390 data citra

jumlah_benar	390
jumlah_file	390
akurasi_pelatihan	100
precision	[1;1;1;1;1;1]
recall	[1;1;1;1;1;1]
f1_score	1

(a)



(b)

Gambar 4.3: (a). Akurasi pelatihan, (b). *Confusion Matrix*

dari Gambar 4.3 (a) menunjukkan akurasi pelatihan yang dilakukan algoritma *K-Nearest Neighbor* dengan $k = 1$ pada 390 data yaitu diperoleh akurasi sebesar 100%, sedangkan Gambar 4.3 (b) menunjukkan *Confusion Matrix* dari hasil pelatihan dengan menggunakan algoritma *K-Nearest Neighbor*, dimana untuk setiap kelas memiliki jumlah benar yang sama antara *true class* dengan *predicted class*.

b. Pelatihan algoritma K-NN dengan penggunaan $k = 1$ pada 138 data citra

Berikut hasil akurasi pelatihan yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 1$ pada 138 data citra

 jumlah_benar	138
 jumlah_file	138
 akurasi_pelatihan	100
 precision	[1;1;1;1;1;1]
 recall	[1;1;1;1;1;1]
 f1_score	1

(a)

		Figure 1							
		File	Edit	View	Insert	Tools	Desktop	Window	Help
True Class	Ekspresi Bahagia	23							
	Ekspresi Jijik		23						
	Ekspresi Marah			23					
	Ekspresi Sedih				23				
	Ekspresi Takut					23			
	Ekspresi Terkejut						23		
		Ekspresi Bahagia	Ekspresi Jijik	Ekspresi Marah	Ekspresi Sedih	Ekspresi Takut	Ekspresi Terkejut		
Predicted Class									

(b)

Gambar 4.4: (a). Akurasi pelatihan, (b). *Confusion Matrix*

dari Gambar 4.4 (a) menunjukkan akurasi pelatihan yang dilakukan algoritma *K-Nearest Neighbor* dengan $k = 1$ pada 138 data diperoleh akurasi sebesar 100%, sedangkan Gambar 4.4 (b) tersebut menunjukkan *Confusion Matrix* dari hasil pelatihan dengan menggunakan algoritma *K-Nearest Neighbor*, dimana untuk setiap kelas memiliki jumlah benar yang sama antara *true class* dengan *predicted class*.

c. Pelatihan algoritma K-NN dengan penggunaan $k = 3$ pada 390 data citra

Berikut hasil akurasi pelatihan yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 3$ pada 390 data citra

 jumlah_benar	182
 jumlah_file	390
 akurasi_pelatihan	46.66
 precision	0.5422
 recall	0.4667
 f1_score	0.5016

(a)

		Figure 1							
		File	Edit	View	Insert	Tools	Desktop	Window	Help
True Class	Ekspresi Bahagia	59	1	1	2			2	
	Ekspresi Jijik	23	38			1	1	2	
	Ekspresi Marah	13	17	29			4	2	
	Ekspresi Sedih	13	20	7	24	1			
	Ekspresi Takut	23	14	10	3	10	5		
	Ekspresi Terkejut	15	15	6	7		22		
		Ekspresi Bahagia	Ekspresi Jijik	Ekspresi Marah	Ekspresi Sedih	Ekspresi Takut	Ekspresi Terkejut		
Predicted Class									

(b)

Gambar 4.5: (a). Akurasi pelatihan, (b). *Confusion Matrix*

dari Gambar 4.5 (a) menunjukkan akurasi pelatihan yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 390 dan penggunaan $k = 3$, dimana diperoleh nilai akurasi sebesar 46%, sedangkan Gambar 4.5 (b) menunjukkan *Confusion Matrix* dari hasil pelatihan dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama.

d. Pelatihan algoritma K-NN dengan penggunaan $k = 3$ pada 138 data citra

Berikut hasil akurasi pelatihan yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 3$ pada 138 data citra

jumlah_benar	59
jumlah_file	138
akurasi_pelatihan	42.7536
precision	0.4691
recall	0.4275
f1_score	0.4473

(a)

True Class	Predicted Class					
	Ekspresi Bahagia	Ekspresi Jijik	Ekspresi Marah	Ekspresi Sedih	Ekspresi Takut	Ekspresi Terkejut
Ekspresi Bahagia	17		1	2		3
Ekspresi Jijik	2	19	1		1	
Ekspresi Marah	6	7	10			
Ekspresi Sedih	9	7	1	5	1	
Ekspresi Takut	7	5		4	6	1
Ekspresi Terkejut	10	4	4	3		2

(b)

Gambar 4.6 : (a). Akurasi pelatihan, (b). Confusion Matrix

dari Gambar 4.6 (a) menunjukkan akurasi pelatihan yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 138 dan penggunaan $k = 3$ adalah sebesar 42.7%, sedangkan Gambar 4.6 (b) tersebut menunjukkan *Confusion Matrix* dari hasil pelatihan dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama, sehingga tingkat akurasinya hanya 42.7%.

e. Pelatihan algoritma K-NN dengan penggunaan $k = 5$ pada 390 data citra

Berikut hasil akurasi pelatihan yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 5$ pada 390 data citra

jumlah_benar	157
jumlah_file	390
akurasi_pelatihan	40.2564
precision	0.3973
recall	0.4026
f1_score	0.3999

(a)

True Class	Predicted Class					
	Ekspresi Bahagia	Ekspresi Jijik	Ekspresi Marah	Ekspresi Sedih	Ekspresi Takut	Ekspresi Terkejut
Ekspresi Bahagia	37	7	5	5	5	6
Ekspresi Jijik	6	30	8	13	5	3
Ekspresi Marah	9	6	31	4	7	8
Ekspresi Sedih	12	11	5	29	6	2
Ekspresi Takut	13	12	11	7	15	7
Ekspresi Terkejut	17	9	10	9	5	15

(b)

Gambar 4.7 : (a). Akurasi pelatihan, (b). Confusion Matrix

dari Gambar 4.7 (a) menunjukkan akurasi pelatihan yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 390 dan penggunaan $k = 5$ adalah sebesar 40%, sedangkan Gambar 4.7 (b) tersebut menunjukkan *Confusion Matrix* dari hasil pelatihan dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama, sehingga tingkat akurasinya hanya 40%.

f. Pelatihan algoritma K-NN dengan penggunaan $k = 5$ pada 138 citra

Berikut hasil akurasi yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 5$ pada 138 data citra

jumlah_benar	53
jumlah_file	138
akurasi_pelatihan	38.4058
precision	0.3999
recall	0.3841
f1_score	0.3918

(a)

True Class	Predicted Class					
	Ekspresi Bahagia	Ekspresi Jijik	Ekspresi Marah	Ekspresi Sedih	Ekspresi Takut	Ekspresi Terkejut
Ekspresi Bahagia	11		3	5		4
Ekspresi Jijik	3	10	3	2	3	2
Ekspresi Marah	8	4	8	1	1	1
Ekspresi Sedih	5	2	2	9	5	
Ekspresi Takut	2	1	2	8	8	2
Ekspresi Terkejut	6	2	2	3	3	7

(b)

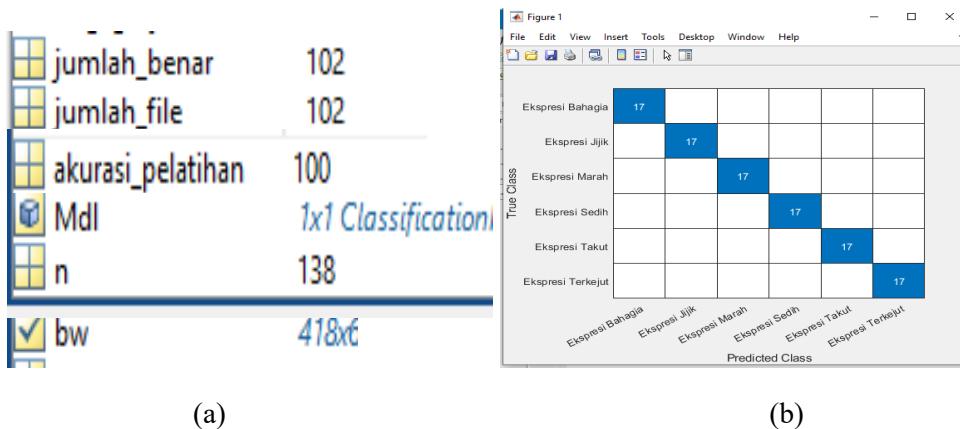
Gambar 4.8 : (a). Akurasi pelatihan, (b). Confusion Matrix

dari Gambar 4.8 (a) menunjukkan akurasi pelatihan yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 138 dan penggunaan $k = 5$. Adalah 38%, sedangkan Gambar 4.16 (b) menunjukkan *Confusion Matrix* dari hasil pelatihan dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama.

2. Pengujian

a. Pengujian algoritma K-NN dengan penggunaan $k = 1$ pada 102 data citra

Berikut hasil akurasi pengujian yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 1$ pada 102 data citra



(a)

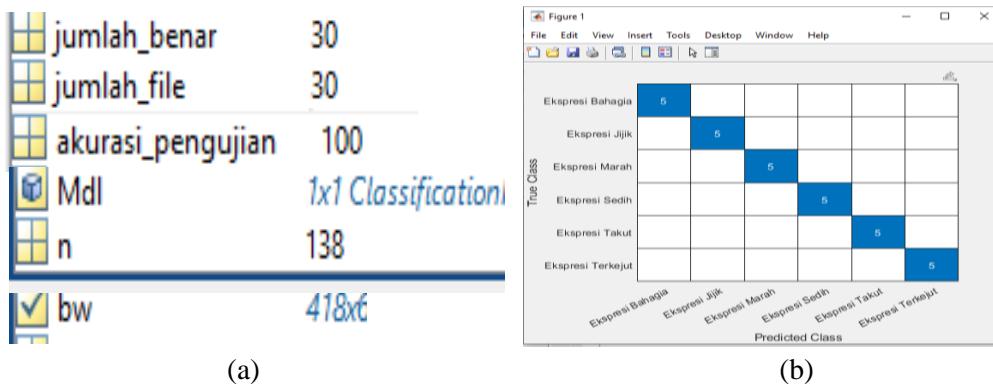
(b)

Gambar 4.9 : (a). Akurasi pengujian, (b). *Confusion Matrix*

dari Gambar 4.9 (a) menunjukkan akurasi pengujian yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 102 data dan penggunaan $k = 1$ adalah 100%, sedangkan Gambar 4.9 (b) menunjukkan *Confusion Matrix* dari hasil pengujian dengan menggunakan algoritma *K-Nearest Neighbor*, dimana untuk setiap kelas memiliki jumlah benar yang sama antara *true class* dengan *predicted class*.

b. Pengujian algoritma K-NN dengan penggunaan $k = 1$ pada 30 data citra

Berikut hasil akurasi pengujian yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 1$ pada 30 data citra.



(a)

(b)

Gambar 4.10 : (a). Akurasi pengujian, (b). *Confusion Matrix*

dari Gambar 4.10 (a) menunjukkan akurasi pengujian yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 30 data dan penggunaan $k = 1$ adalah 100%, sedangkan Gambar 4.10 (b) menunjukkan *Confusion Matrix* dari hasil pengujian dengan menggunakan algoritma *K-Nearest Neighbor*, dimana untuk setiap kelas memiliki jumlah benar yang sama antara *true class* dengan *predicted class*.

c. Pengujian algoritma K-NN dengan penggunaan $k = 3$ pada 102 citra

Berikut hasil akurasi pengujian yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 3$ pada 102 data citra



(a)

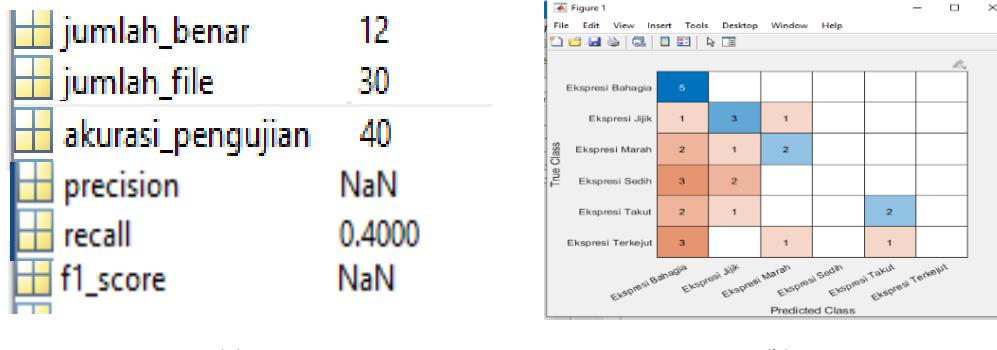
(b)

Gambar 4.11 : (a). Akurasi pengujian, (b). *Confusion Matrix*

dari Gambar 4.11 (a) menunjukkan akurasi pengujian yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 102 data dan penggunaan $k = 3$ adalah sebesar 40%, sedangkan Gambar 4.11 (b) tersebut menunjukkan *Confusion Matrix* dari hasil pengujian dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama, sehingga tingkat akurasinya hanya 40%.

d. Pengujian algoritma K-NN dengan penggunaan $k = 3$ pada 30 citra

Berikut hasil akurasi pengujian yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 3$ pada 30 data citra



(a)

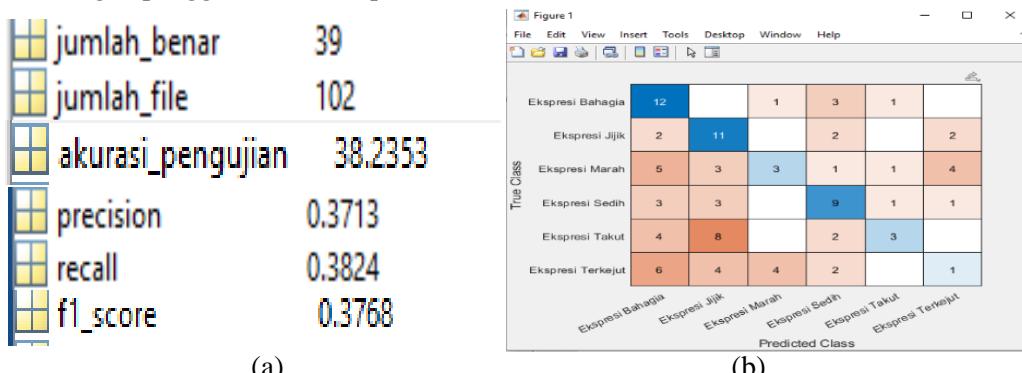
(b)

Gambar 4.12 : (a). Akurasi pengujian, (b). *Confusion Matrix*

dari Gambar 4.12 (a) menunjukkan akurasi pengujian yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 30 data dan penggunaan $k = 3$ adalah 40%, sedangkan Gambar 4.11 (b) tersebut menunjukkan *Confusion Matrix* dari hasil pengujian dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama, sehingga tingkat akurasinya hanya 40%.

e. Pengujian algoritma K-NN dengan penggunaan $k = 5$ pada 102 citra

Berikut hasil akurasi pengujian yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 5$ pada 102 data citra



(a)

(b)

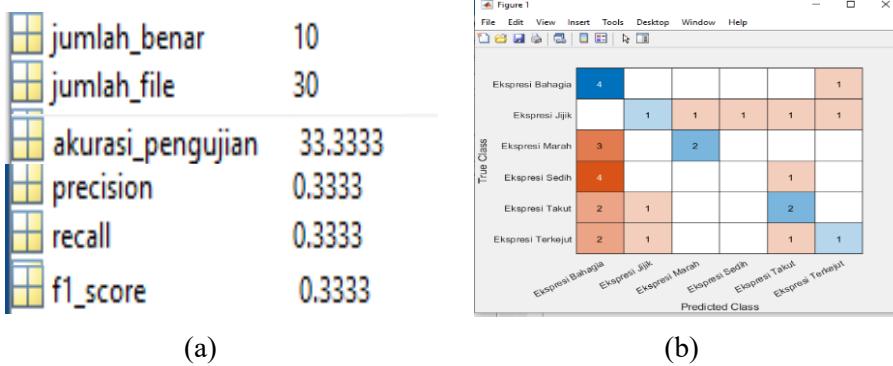
Gambar 4.13 : (a). Akurasi pengujian, (b). *Confusion Matrix*

dari Gambar 4.13 (a) menunjukkan akurasi pengujian yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 102 data dan penggunaan $k = 5$ adalah 38%, sedangkan Gambar 4.13 (b) tersebut menunjukkan *Confusion Matrix* dari hasil pengujian dengan menggunakan algoritma *K-Nearest*

Neighbor, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama, sehingga tingkat akurasinya hanya 38%.

f. Pengujian algoritma K-NN dengan penggunaan $k = 5$ pada 30 citra

Berikut hasil akurasi pengujian yang diperoleh dari tampilan Matlab pada implementasi algoritma K-NN dengan penggunaan $k = 5$ pada 30 data citra



Gambar 4.14 : (a). Akurasi pengujian, (b). *Confusion Matrix*

dari Gambar 4.14 (a) menunjukkan akurasi pengujian yang dilakukan algoritma *K-Nearest Neighbor* dengan jumlah data sebanyak 30 data dan penggunaan $k = 5$. Hasil tersebut menunjukkan nilai akurasi sebesar 33%, sedangkan Gambar 4.14 (b) menunjukkan *Confusion Matrix* dari hasil pengujian dengan menggunakan algoritma *K-Nearest Neighbor*, dimana antara *true class* dengan *predicted class* tidak memiliki jumlah benar yang sama, sehingga tingkat akurasinya hanya 33%.

5. KESIMPULAN

Setelah melakukan perancangan, implementasi dan pengujian, kesimpulan yang didapatkan pada penelitian ini adalah sebagai berikut:

1. Penerapan Algoritma *K-Nearest Neighbor* (K-NN) dalam mengklasifikasikan berbagai jenis ekspresi wajah manusia dilakukan dengan membandingkan jumlah data yang diproses, yaitu data dengan jumlah yang lebih banyak dan data dengan jumlah yang lebih sedikit. Berdasarkan hasil pengujian, data yang lebih banyak menghasilkan tingkat akurasi yang lebih tinggi.
2. Kinerja algoritma *K-Nearest Neighbor* dengan ekstraksi ciri GLCM menghasilkan beberapa akurasi sesuai penggunaan ketetapanan (k). Akurasi tertinggi yang didapatkan adalah sebesar 100% dengan penentuan $k = 1$.

DAFTAR PUSTAKA

- Abidin, Z. 2012. Rancang Bangun Sistem Pengenalan Ekspresi Wajah Menggunakan *Fisherface* dan Jaringan Syaraf Tiruan Backpropagation. *Jurnal MIPA*, 35(02) : 194–203.
- Ahmad, N., & Alhamad, A. 2019. Penerapan Metode GLCM (*Gray Level Co-Occurrence Matrix*) untuk Reduksi Ciri Pada Pengenalan Ekspresi Wajah. *Jurnal Cosphi*, 3(2) : 61–64.
- Ali, R. 2016. Detektor Ekspresi Wajah Manusia. *Jurnal Informatika*, 16(1): 74–84.
- Andono, P.N., T.Sutojo., & Muljono. 2017. **Pengolahan Citra Digital**. Penerbit ANDI. Yogyakarta. Hal. 24-75.
- Asmara, R. A. ., Andjani, B. S. ., Rosiani, U. D. ., & Choirina, P. . (2018). KLASIFIKASI JENIS KELAMIN PADA CITRA WAJAH MENGGUNAKAN METODE NAIVE BAYES. *Jurnal Informatika Polinema*, 4(3), 212–217. <https://doi.org/10.33795/jip.v4i3.209>
- Hamdi, AS., & Bahrudin, E. 2014. **Metode Penelitian Kuantitatif Aplikasi dalam Pendidikan**. Deepublish. Yogyakarta. Hal. 5-38.
- ihsan, mohamad, Niswatin, R. K., & Swanjaya, D. (2021). DETEKSI EKSPRESI WAJAH MENGGUNAKAN TENSORFLOW. *Joutica*, 6(1), 428–433. <https://doi.org/10.30736/jti.v6i1.554>
- Ilimageo, M. A. Z. 2020. Implementasi Metode K-Nearest Neighborhood untuk Deteksi Emosi Pada Wajah. *Skripsi*. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Malang.
- Kusumawati, A. 2020. Implementasi Local Binary Pattern dan K-Nearest Neighbor Untuk Klasifikasi Citra Wajah. *Skripsi*. Program Studi Informatika Fakultas Teknologi Informasi dan Elektro Universitas Teknologi Yogyakarta. Yogyakarta.

- Mahran, A. A., Hapsari, R. K., & Nugroho, H. 2020. Penerapan *Naive Bayes Gaussian* Pada Klasifikasi Jenis Jamur Berdasarkan Ciri Statistik Orde Pertama. *Jurnal Ilmiah NERO*. 5(2) : 91-99. <https://doi.org/10.21107/nero.v5i2.165>
- Manuel, I. S., & Ernawati, I. 2020. Implementasi GLCM dan Algoritma *Naive Bayes* Dalam Klasifikasi Jenis Bunga Anggrek. *Senamika*, 1(2) : 99–109.
- Nendya, M. B., Yuniarno, E. M., & Sumpeno, S. (2021). Clustering Titik Fitur Model Wajah 3D Menggunakan K-Nearest Neighbour. *Jurnal Informatika Dan Sistem Informasi*, 7(1), 19–24. Retrieved from <https://jurnal.uc.ac.id/index.php/JUISI/article/view/1739>
- Putry, N. M., Sary, B.N. 2022. Komparasi Algoritma KNN Dan *Naïve Bayes* Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus. *EVOLUSI: Jurnal Sains Dan Manajemen*, 10(1) : 45-57. <https://doi.org/10.31294/evolusi.v10i1.12514>
- Reynaldo, R. R., & Maliki, I. 2021. Pengenalan Ekspresi Wajah dengan Metode *Viola Jones* dan *Convolutional Neural Network*. *Komputika : Jurnal Sistem Komputer*, 10(1) :1–9. <https://doi.org/10.34010/KOMPUTIKA.V10I1.4119>
- Seandrio, A. L., Pratomo, A. H., & Florestiyanto, M. Y. 2021. Implementation of Convolutional Neural Network (CNN) in Facial Expression Recognition Implementasi Convolutional Neural Network (CNN) Pada Pengenalan Ekspresi Wajah. *Jurnal Informatika Dan Teknologi Informasi*, 18(2) : 211–221. <https://doi.org/10.31315/telematika.v18i2.4823>
- Syahid, M. A. 2021. Pengenalan Wajah Berbasis Fitur Principal Component Analysis (PCA) Dengan Seleksi Fitur Genetic Algorithm (GA) Dan K – Nearest Neighbor (K-NN) Classifier. *skripsi*. Program Studi Teknik Informatika. Fakultas Teknik Universitas Mataram. Mataram.
- Zana, A. Z. B., Raharjo, J., & Fauzi, H. 2021. Analisa Jenis Kelamin Berdasarkan Citra Wajah Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) Dan Klasifikasi *Naive Bayes* *Gender Analysis Based on Face Image Using Gray Level Co-Occurrence Matrix* (GLCM) and *Naive Bayes Classification*. *E-Proceeding of Engineering*. 8(5) : 4580–4591.