

# Mendeteksi Ekspresi Wajah dengan Meninjau Iris Mata Menggunakan Metode Transformasi Hough dan K-Nearest Neighbor (KNN)

Ningsih Djamsi<sup>1</sup>, Didih Rizki Chandranegara<sup>2</sup>, Zamah Sari<sup>3</sup>

<sup>1,2,3</sup>Universitas Muhammadiyah Malang

ningsihdjamsi@webmail.umm.ac.id<sup>1</sup>, didihrizki@umm.ac.id<sup>2</sup>, zamahsari@umm.ac.id<sup>3</sup>

## Abstrak

Wajah memiliki perbedaan (ciri khas), selain itu wajah juga merupakan pusat dari ekspresi pada manusia. Secara tidak langsung ekspresi wajah menjadi suatu komunikasi non-verbal, banyak faktor-faktor yang dapat mendukung untuk mendeteksi ekspresi wajah seseorang contohnya seperti melihat pergerakan dan perubahan pada alis, bibir dan mata. Dalam penelitian ini lebih berfokus pada iris mata sebagai objek pengamatan. Terdapat dua skenario pengujian untuk mendeteksi ekspresi wajah berdasarkan citra wajah, dengan menggunakan dataset TFEIDHigh. Skenario pertama menggunakan metode K-Nearest Neighbor (K-NN) dengan menghasilkan akurasi sebesar 80.0% dan waktu eksekusi 61.3831 Seconds. Skenario kedua menggunakan metode Convolutional Neural Network (CNN) dengan menghasilkan akurasi sebesar 80.0% dengan waktu eksekusi 656.730 Seconds.

**Kata Kunci:** Ekspresi Wajah, iris mata, Klasifikasi, Transformasi Hough K-Nearest Neighbor

## Abstract

The face has differences (characteristics), besides that the face is also the center of expression in humans. Directly facial expressions become a non-verbal communication, many factors can support detecting a person's facial expressions such as movements and changes in eyebrows, lips and eyes. In this study, the focus is on the iris as the object of observation. There are two test scenarios to detect facial expressions based on facial images, using the TFEIDHigh dataset. The first scenario uses the KNearest Neighbor (K-NN) method with an accuracy quality of 80.0% and an execution time of 61.3831 Seconds. The second scenario uses the Convolutional Neural Network (CNN) method with an accuracy of 80.0% with an execution time of 656,730 Seconds.

**Keywords:** Facial Expression, Iris, Classification, Hough K-Nearest Neighbor Transformation

## 1. Pendahuluan

Wajah adalah salah satu bagian dari tubuh manusia yang paling sering menjadi pusat perhatian karena wajah terdiri dari empat organ penting yaitu mata, hidung, mulut, lidah dan telinga. Struktur wajah juga berbeda-beda tiap manusia yang paling jelas terlihat biasanya dari bentuk wajah, hidung, mulut, dan mata [1]. maka dari itu biasanya cara mengenali seseorang adalah dengan melihat wajahnya karna disetiap wajah orang memiliki perbedaan (ciri khas), selain itu wajah juga merupakan pusat dari ekspresi pada manusia. Pada dasarnya manusia menggunakan ekspresi wajah sebagai komunikasi untuk menggambarkan isi hati atau perasaan yang sedang dirasakan, masing-masing individu memiliki cirinya sendiri dalam mengekspresikan perasaannya.

Perasaan seperti marah, senang, sedih atau pun takut dapat terlihat jelas pada ekspresi wajah seseorang, secara tidak langsung ekspresi wajah menjadi suatu komunikasi yang berbentuk isyarat dalam kehidupan sehari-hari. Komunikasi non-verbal ini dapat memudahkan kita dalam berkomunikasi antar sesama manusia, banyak faktor-faktor yang dapat mendukung untuk mendeteksi ekspresi wajah seseorang contohnya seperti melihat pergerakan dan perubahan pada alis, bibir dan mata [2] mata memiliki sifat yang unik sehingga mempunyai pola yang berbeda. [3]. Mendeteksi ekspresi wajah seseorang bisa dilihat dari pupil mata. Pupil mata terletak ditengah iris (bagian yang berwarna pada mata) [4]. iris mata adalah organ yang dapat terlihat berbentuk lingkaran yang mengelilingi pupil mata, Analisa ini berfokus pada perubahan yang terjadi pada bola mata agar dapat mengetahui perasaan seseorang. fungsi iris mata sendiri

ialah mengatur intensitas cahaya yang masuk kedalam mata dengan membesar dan mengecilkan pupil mata.

Dalam penelitian ini lebih berfokus pada iris mata sebagai objek pengamatan, karna dari hasil penelitaian sebelumnya yang menggunakan wajah sebagai objek pengamatan didapatkan akurasi yang rendah seperti pada penelitian yang dilakukan oleh Derry Alamsyah,dkk [5] dengan judul *implementasi convolutional neural networks(CNN) untuk klasifikasi ekspresi citra wajah pada FER-2013 dataset* dengan hasil akurasi sebesar 66%. Berdasarkan hal tersebut maka dijadikanlah iris mata sebagai objek pengamatan, agar identifikasi lebih berfokus pada satu area pada wajah. Penelitian ini mengimplementasi metode Transformasi Hough yang merupakan metode *image processing* untuk mendeteksi lingkaran dan garis pada citra. Beberapa tahun terakhir, penelitian mengenai identifikasi iris mata makin pesat seperti pada penelitian sebelumnya yang dilakukan Arjon Samuel Sitio,dkk [3] dengan judul *Sistem Identifikasi Biometrik Ekpresi Wajah Menggunakan Metode Transformasi*

*Hough* dengan keberhasilan pendektasian 8085%.

Selain metode transformasi hough dalam penelitian ini juga menggunakan KNearest Neighbor (KNN) untuk mengklasifikasi hasil dataset dari processing menggunakan transformasi hough [6]. K-Nearest Neighbor (KNN) merupakan metode klasifikasi berdasarkan meninjau objek atau tetangga terdekat. Penelitian sebelumnya yang dilakukan beberapa tahun terakhir ini ialah oleh Rifki Kosasih [1] dengan judul *Kombinasi metode isomap dan KNN pada image processing untuk pengenalan wajah* dengan menggunakan metode isomap dan hasil akurasi 83,33%.

Studi ini mengusulkan metode *transformasi hough* untuk mendeteksi ekspresi wajah dan menggunakan *K-Nearest Neighbor (KNN)* untuk klasifikasi hasil dari processing metode *transformasi hough*. Perbedaan penelitian ini dengan penelitian sebelumnya ialah pada penggunaan *K-Nearest Neighbor (KNN)* diharapkan dapat meningkatkan keberhasilan deteksi wajah.

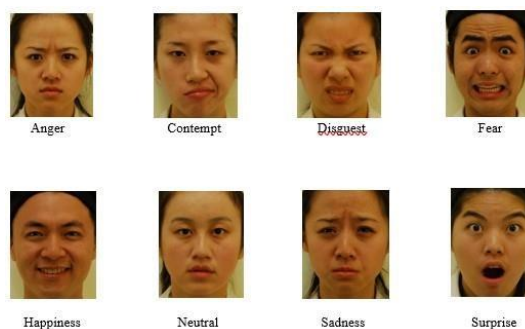
## 2. Metode Penelitian

### 2.1 Dataset

Dataset yang dipakai dalam penelitian ini adalah TFEIDHigh. Didalam dataset yang digunakan ini terdapat gambar ekspresi wajah yang terdiri dari 8 kategori dengan ekspresi wajah yang berbeda-beda. Contoh kategorinya seperti pada Tabel 1 dan Gambar 2.

Kelas	Jumlah Data
Anger	34
Contempt	68
Disgust	40
Fear	40
Happiness	40
Neutral	39
Sadness	39
Surprise	36

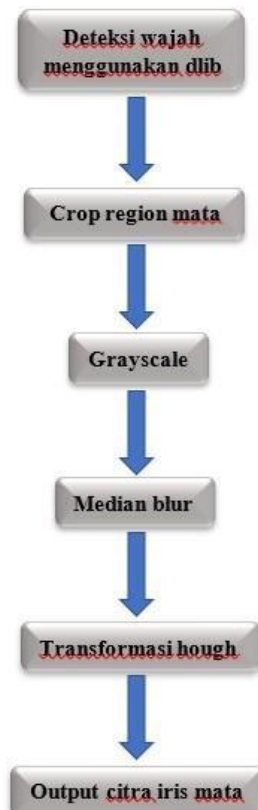
Tabel 1. Pembagian Kelas di Dataset TFEIDHigh



Gambar 1. Contoh gambar masing-masing kelas di dataset TFEIDHigh

## 2.2 Preprocessing

Pada tahapan preprocessing ini merupakan tahapan meningkatkan citra agar mempermudah analisa pengerjaan selanjutnya. Ada pula tahapan preprocessing yang dilakukan seperti pada Gambar 2.



Gambar 2. Proses preprocessing

## 2.3 Pengujian

Pada k-Nearest Neighbour (k-NN) menyimpan instance pada satu tempat yang sama,  $n$  merupakan fitur instance yang telah di definisikan sebelumnya.[15] Matrik distance biasanya digunakan untuk mengukur jarak antar instance,[16] dalam pengukuran jarak dipakai perhitungan Euclidean distance pada Persamaan 1 adalah perhitungan jarak yang paling sering digunakan dalam algoritma KNN.

Keterangan :  $d(x,y)$  = jarak antara data  $x$  ke data  $y$   $x_i$  = data testing ke- $i$   $y_i$  = data training ke- $i$   $n$  = dimensi data.

$$d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

## 2.4 Evaluasi

Penelitian ini menggunakan Classification Report untuk mengukur performa sistem yang dihitung menggunakan rumus *accuracy*, *precision*, *recall*, dan *F1-score*.

1) *Accuracy*, merupakan rasio prediksi benar terhadap keseluruhan data untuk mengevaluasi kinerja model klasifikasi. Rumus perhitungan *accuracy* ditulis dengan Persamaan 2.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (2)$$

2) *Precision*, merupakan proses evaluasi prediksi benar positif terhadap keseluruhan data oleh sistem. Rumus perhitungan *precision* ditulis dengan Persamaan 3.

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

3) *Recall*, merupakan nilai yang melambangkan seberapa baik hasil model terlatih untuk mengenali data yang signifikan. Rumus perhitungan *recall* ditulis dengan Persamaan 3.

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

4) *F1-score* merupakan perbandingan rata-rata *precision* dan *recall*. Rumus perhitungan *F1-score* ditulis dengan Persamaan 4.

$$F1 = 2 \times \frac{Precision \times recall}{Precision + recall} \quad (4)$$

## 2.5 Confusion Matrix

Suatu metode untuk menghitung ketepatan data sehingga data dapat digunakan dalam sistem pendukung keputusan adalah Confusion matrix. Gambaran dari confusion matrix ditunjukkan oleh Gambar 4.

		Predicted Labels	
		TP (True Positive)	FP (False Positive)
True Labels	FN (False Negative)		
	TN (True Negative)		

Gambar 4. Confusion Matrix

Berdasarkan Gambar 4 terdapat empat istilah dalam confusion matrix. Keempat istilah tersebut adalah:

1. True Positive adalah data positive yang diprediksi dengan benar.
2. True Negative adalah data negative yang diprediksi dengan benar.
3. False Positive adalah data negative yang diprediksi sebagai positive.
4. False Negative adalah data positive yang diprediksi sebagai negative

## 3. Hasil dan Kesimpulan

Dalam tahapan ini terdapat sejumlah skenario pengujian. Sebelum tahapan pengujian dilakukan, skenario pengujian terlebih dahulu dilakukan. Proses preprocessing menggunakan Transformasi Hough untuk mendeteksi garis dan lingkaran pada citra mata, selain itu dalam preprocessing ini menggunakan crop region mata, grayscale, dan median blur. Hasil preprocessing tadi akan diimplementasikan untuk metode dalam melakukan skenario pengujian. Berikut merupakan detail skenario pengujian yang akan dijalankan:

- a) Skenario pertama dilakukan pelatihan metode K-Nearest Neighbour (K-NN) pada dataset TFEIDHigh dengan parameter yang telah didapat dari proses preprocessing pada Gambar 5 dan Gambar 6.

```
Akurasi KNN : 80.0 %
All done in 61.38415789604187 seconds ---
```

Gambar 5. Hasil Akurasi

```

# KNN Predictor
benar = 0
k = 7
for i in range(len(test_images)):
    print("test_images[i]", test_images[i])
    euclidean = np.sqrt(np.sum(np.power((train_images - test_images[i]), 2), axis=1))

    k_label = train_labels[euclidean.argsort()][:k]
    (unique, counts) = np.unique(k_label, return_counts=True)
    print("k :", k, k_label)
    print(np.asarray((unique, counts)).T)
    prediksi = unique[np.argmax(counts)]
    print("Prediksi :", prediksi)
    print("Real class :", test_labels[i], "\n")
    if prediksi == test_labels[i]:
        benar += 1
print("Akurasi :", benar / len(test_labels) * 100, "%")
print("\nAll done in %s seconds ---" % (time.time() - start_time))

```

Gambar 6. Implementasi KNN

b) Skenario kedua dilakukan pelatihan metode

Convolutional Neural Network (CNN) pada dataset TFEIDHigh dengan parameter yang telah didapat dari proses preprocessing Gambar 7 dan Gambar 8.

```

class = np.unique(np.array([labelencoding[i] for i in labels]))
num_classes = len(labelencoding[labelencoding.index(class)]/num_classes)
print("num_classes :", num_classes)
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dense(10, activation='softmax'))
model.add(tf.keras.layers.Dense(10, activation='softmax'))
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])
model.summary()
model.fit(np.array(X), np.array(y), batch_size=64, epochs=100, verbose=1, validation_data=(np.array(test), np.array(y_test)))

test_loss, test_acc = model.evaluate(np.array(test), np.array(y_test), verbose=2)
print("\nAkurasi CNN :", test_acc)
print("\nAkurasi CNN :", test_acc * 100, "%")
print("\nAll done in %s seconds ---" % (time.time() - start_time))

```

Gambar 7. Implementasi CNN

Akurasi CNN : 80.0000011920929 %

All done in 656.7304358482361 seconds ---

Gambar 8. Hasil Akurasi

Dapat dilihat akurasi yang diperoleh dari dua metode Convolutional Neural Network dan K-Nearest Neighbour (KNN) sama dengan hasil akurasi sebesar 80,0%. Dikarenakan tahapan preprocessing yang dilalui sama- sama menggunakan metode Transformasi Hough. Namun terdapat perbedaan pada waktu eksekusi dimana terlihat metode K-Nearest Neighbour (K-NN) memiliki Waktu eksekusi 61.3831 Seconds jauh lebih baik dari metode Convolutional Neural Network yang hanya 656.730 Seconds.

#### 4. Kesimpulan

Berdasarkan pengujian yang telah dilakukan, metode yang diusulkan dalam penelitian ini dapat meningkatkan performa metode penelitian sebelumnya yang hanya menggunakan transformasi hough. Dalam pengujian ini juga menambahkan K-nearest neighbors (KNN) untuk mengklasifikasi dataset yang telah di proses menggunakan transformasi hough sehingga akan menaikkan nilai akurasi. Pada penelitian ini menerapkan dua skenario dengan dataset dan tahapan preprocessing yang sama. Dari skenario yang diterapkan didapatkan hasil bahwa skenario satu dan dua memiliki nilai akurasi yang sama, tetapi skenario pertama yang

menggunakan K-Nearest Neighbor (KNN) memiliki waktu eksekusi lebih baik dibandingkan dengan skenario kedua yang menggunakan Convolutional Neural Network.

### Referensi

- [1] R. Kosasih, "Kombinasi Metode ISOMAP Dan KNN Pada Image Processing Untuk Pengenalan Wajah," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 5, no. 2, p. 166, 2020, doi: 10.24114/cess.v5i2.18982.
- [2] N. A. Mayangky, R. Rousyati, D. Riana, S. Hadianti, R. Nurfalah, and M. R. Kusumayudha, "Analisa Perbandingan Metode Histogram Dan Median Filtering Pada Citra Mata," *J. Responsif Ris. Sains dan Inform.*, vol. 3, no. 2, pp. 161–165, 2021, doi: 10.51977/jti.v3i2.505.
- [3] A. Sindar and A. S. Sitio, "Sistem Identifikasi Biometrik Ekspresi Wajah Menggunakan Metode Transformasi Hough," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 3, no. 3, pp. 262–267, 2020, doi: 10.32672/jnkti.v3i3.2722.
- [4] M. S. Purba, "Perancangan Sistem Identifikasi Biometrik Iris Mata Menggunakan Metode Transformasi Hough," *J. Maj. Ilm. Inf. dan Teknol. Ilm.*, vol. 7, no. 2, pp. 117–122, 2020.
- [5] D. Alamsyah and D. Pratama, "Implementasi Convolutional Neural Networks (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada FER-2013 Dataset," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 350–355, 2020, doi: 10.36294/jurti.v4i2.1714.
- [6] F. Fandiansyah, J. Y. Sari, and I. P. Ningrum, "Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor," *J. Ultim.*, vol. 9, no. 1, pp. 1–9, 2017, doi: 10.31937/ti.v9i1.557.
- [7] A. Akbar and A. Lawi, "Implementasi Algoritma Deep Artificial Neural Network Menggunakan Mel Frequency Cepstrum Coefficient Untuk Klasifikasi Audio Emosi Manusia," pp. 66–73, 2021.
- [8] A. Singh, A. P. Srivastav, P. Choudhary, and S. Raj, "Facial emotion recognition using convolutional neural network," *Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. ICIEM 2021*, pp. 486–490, 2021, doi: 10.1109/ICIEM51511.2021.9445346.
- [9] F. J. Pontoh, F. Xaverius Senduk, and I. E. G. Pondaag, "Aplikasi Pengenalan Iris Mata Menggunakan Metode Hough Transform Dan Gabor Wavelet," *J. Ilm. Inform.*, vol. 9, no. 02, pp. 105–109, 2021, doi: 10.33884/jif.v9i02.4205.
- [10] F. E. Alfian, I. G. P. S. Wijaya, and F. Bimantoro, "Identifikasi Iris Mata Menggunakan Metode Wavelet Daubechies dan K-Nearest Neighbor," *J. Teknol. Informasi, Komputer, dan Apl. (JTika )*, vol. 2, no. 1, pp. 1–10, 2020, doi: 10.29303/jtika.v2i1.76.
- [11] B. C. Wibowo, F. Nugraha, and A. P. Utomo, "Uji Deteksi Objek Bentuk Bola Dengan Menerapkan Metode Circular Hough Transform," *J. Inform. Upgris*, vol. 7, no. 1, 2021, doi: 10.26877/jiu.v7i1.8309.
- [12] A. R. Mubarrok, Haryanto, and D. Rahmawati, "Rancang Bangun Timbangan Buah Anggur Digital Otomatis Berbasis Webcam Menggunakan Transformasi HOUGH," *Sci. Electro*, vol. 12, no. 2, pp. 44–50, 2020.
- [13] C. Wijaya, H. Irsyad, and W. Widhiarso, "Klasifikasi Pneumonia Menggunakan Metode K-Nearest Neighbor Dengan Ekstraksi Glcm," *J. Algoritm.*, vol. 1, no. 1, pp. 33–44, 2020, doi: 10.35957/algoritme.v1i1.431.
- [14] R. Kosasih, "Penggunaan Metode Linear Discriminant Analysis Untuk Pengenalan Wajah Dengan Membandingkan Banyaknya Data Latih," *J. Ilm. Teknol. dan Rekayasa*, vol. 26, no. 1, pp. 25–34, 2021, doi: 10.35760/tr.2021.v26i1.3520.
- [15] P. D. Silitonga and R. Damanik, "Perbandingan Algoritma k-Nearest Neighbors (k-NN) dan Support Vector Machines (SVM) untuk Klasifikasi Pengenalan Citra Wajah," *J. ICT Inf. Commun. Technol.*, vol. 20, no. 1, pp. 186–191, 2021, doi: 10.36054/jictikmi.v20i1.354.
- [16] N. Wakhidah, "Perbaikan Kualitas Citra Menggunakan Metode Contrast Stretching," *J. Transform.*, vol. 8, no. 2, p. 78, 2011, doi: 10.26623/transformatika.v8i2.48.
- [17] J. I. Komputasi, V. No, M. Ssd, V. Mobilenet, and S. Model, "Abstrak," vol. 19, no. September, pp. 421–430, 2020