

# Klasifikasi Citra Aksara Lontara menggunakan K-NN dan Ekstraksi Fitur HOG

Rayhan Saneval Arhinza<sup>1</sup>, Anggraini Puspita Sari<sup>2</sup>, Fawwaz Ali Akbar<sup>3</sup>

<sup>1,2,3</sup> Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur

E-mail: <sup>1</sup>[20081010126@student.upnjatim.ac.id](mailto:20081010126@student.upnjatim.ac.id), <sup>2</sup>[anggraini.puspita.if@upnjatim.ac.id](mailto:anggraini.puspita.if@upnjatim.ac.id),

<sup>3</sup>[fawwaz.ali.if@upnjatim.ac.id](mailto:fawwaz.ali.if@upnjatim.ac.id)

**Corresponden Author:** [anggraini.puspita.if@upnjatim.ac.id](mailto:anggraini.puspita.if@upnjatim.ac.id)

Diterima Redaksi: 16 Mei 2024 Revisi Akhir: 29 Juli 2024 Diterbitkan Online: 11 September 2024

**Abstrak** – Indonesia memiliki kekayaan budaya yang beragam, salah satunya adalah Bahasa daerah yang memiliki tulisan unik di wilayah Nusantara. Salah satu contohnya adalah Aksara Lontara, yang digunakan oleh masyarakat Bugis dan Makasar. Aksara Lontara merupakan salah satu aksara yang ada di Indonesia yang terancam punah karena perkembangan zaman. Seperti pada penelitian yang dilakukan oleh (Fitrawahyudi & Kasmawati, 2019) di Kabupaten Maros dan mendapatkan hasil penelitian bahwa 89% responden tidak mengerti sama sekali Aksara Lontara. Algoritma K-Nearest Neighbors (K-NN) bisa menjadi alat yang dapat digunakan untuk mengenali pola pada Aksara Lontara. Prinsip K-NN cukup sederhana yakni mencocokkan kemiripan data baru dengan data uji terdekat. Pada penelitian ini menggunakan K-NN sebagai klasifikasi dan Histogram of Oriented Gradients (HOG) sebagai ekstraksi fiturnya. Berdasarkan hasil penelitian yang telah dilakukan menggunakan skema pengujian ukuran citra 32×32 pixel, splitting dataset 90:10, dan nilai k=3 mendapatkan hasil akurasi sebesar 0.9235

**Kata Kunci** — K-NN, Histogram of Oriented Gradients, Aksara Lontara

**Abstract** – Indonesia has a rich and diverse cultural heritage, one aspect of which is regional languages with unique scripts found throughout the archipelago. One example is the Lontara script, used by the Bugis and Makassar communities. The Lontara script is one of the scripts in Indonesia that is endangered due to the passage of time. As shown in research conducted by Fitrawahyudi & Kasmawati (2019) in Maros Regency, 89% of respondents did not understand the Lontara script at all. The K-Nearest Neighbors (K-NN) algorithm can be a tool used to recognize patterns in the Lontara script. The principle of K-NN is quite simple: it matches the similarity of new data with the nearest test data. This study uses K-NN for classification and Histogram of Oriented Gradients (HOG) for feature extraction. Based on the research conducted using a test scheme with an image size of 32×32 pixels, a dataset splitting ratio of 90:10, and a k-value of 3, an accuracy result of 0.9235 was achieved.

**Keywords** — K-NN, Histogram of Oriented Gradients, Aksara Lontara



## 1. PENDAHULUAN

Indonesia merupakan salah satu negara yang memiliki kekayaan budaya yang beraneka ragam. Salah satu contohnya adalah bahasa daerah yang memiliki tulisan unik di wilayah nusantara. Aksara Lontara merupakan salah satu aksara yang digunakan oleh masyarakat Bugis dan Makassar. Nama “Lontara” berasal dari nama tumbuhan endemik yang ada di Sulawesi Selatan. Aksara Lontara dikembangkan sejak abad ke-14 oleh penguasa Kerajaan Gowa-Tallo. Aksara Lontara masih digunakan hingga saat ini dalam kegiatan tradisional dan upacara adat. Akan tetapi keberlangsungannya bergantung pada dukungan masyarakat, terutama dari kalangan Bugis dan Makassar agar tidak tergerus oleh modernisasi karena Aksara Lontara merupakan salah satu aksara yang ada di Indonesia yang kini terancam Punah. Berbeda dari aksara Jawa yang masih cukup populer di Indonesia, Aksara Lontara jarang dikenal oleh masyarakat umum. Hal tersebut menunjukkan bahwa mayoritas masyarakat tidak akrab dengan Aksara Lontara yang merupakan Sebagian dari bahasa daerah Bugis.

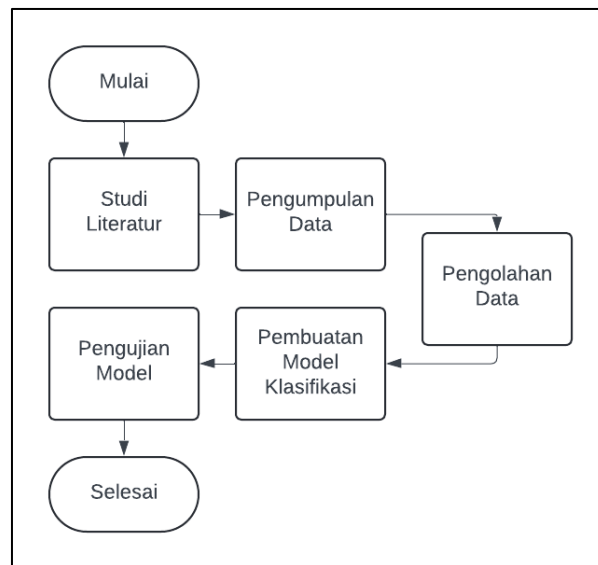
Tidak hanya itu penelitian mengenai Aksara Lontara pun terbilang masih cukup sedikit dibandingkan dengan Aksara Jawa. Oleh karena itu, penelitian ini dilakukan agar dapat dijadikan acuan untuk penelitian yang akan datang. Aksara bukanlah milik satu bidang ilmu tertentu saja, sehingga semua akademisi dan peneliti dapat berkontribusi dalam mempelajari aksara, mengingat setiap orang memiliki kemampuan dan perspektif yang berbeda-beda (Sinaga, 2023). Pada penelitian yang dilakukan oleh (Fitrawahyudi & Kasmawati, 2019) di Kabupaten Maros menggunakan metode kuesioner mendapatkan hasil penelitian bahwa hanya 2.5% responden

atau sekitar 5 orang yang mampu membaca dan menulis dengan benar, 8,5% atau sekitar 17 orang , dan 89% atau sekitar 188 orang tidak bisa sama sekali membaca dan menulis Aksara Lontara ini. Dari hasil penelitian tersebut menunjukkan bahwa semakin sedikitnya masyarakat yang mengenal Aksara Lontara ini

Dengan perkembangan teknologi saat ini, berbagai pendekatan dapat digunakan untuk membantu Masyarakat untuk mengetahui pola Aksara Lontara. Salah satu pendekatan yang saat ini semakin populer yakni bidang machine learning. Pendekatan ini dapat memungkinkan mesin untuk meniru kemampuan manusia dalam mengenali pola. Pada machine learning terdapat beberapa metode yang dapat digunakan untuk klasifikasi seperti deep learning dan algoritma dasar. CNN merupakan salah satu contoh deep learning yang saat ini seringkali digunakan. CNN adalah jenis algoritma deep learning yang didasarkan pada feed forward neural network (FCNN). CNN dapat diterapkan pada beberapa dimensi seperti 1D hingga 4D [1]. sedangkan algoritma dasar yang kini seringkali digunakan adalah K-NN. KNN adalah salah satu jenis algoritma supervised learning yang seringkali digunakan untuk mendeteksi objek baru berdasarkan tetangga terdekatnya [2]. Metode yang digunakan pada penelitian ini adalah HOG sebagai ekstraksi fiturnya dan K-NN sebagai klasifikasinya. Pada penelitian yang dilakukan oleh (Widiarti & Suparwito, 2022) mengenai klasifikasi citra Aksara Bali menggunakan metode K-NN dan ekstraksi fitur *intensity of pixel* mendapatkan hasil akurasi sekitar 0.8681. Pada penelitian Aksara Lontara ini akan menggunakan metode K-NN karena K-NN merupakan metode yang cukup simple namun dapat menghasilkan akurasi yang cukup tinggi, tidak hanya itu K-NN juga cukup cocok untuk dipadukan dengan output dari ekstraksi fitur HOG.

## 2. METODE PENELITIAN

Pada Penelitian ini menggunakan metode *Histogram of Oriented Gradients* (HOG) sebagai ekstraksi fiturnya, KNN sebagai klasifikasinya, dan dataset citra Aksara Lontara. terdapat beberapa tahapan alur penelitian dijelaskan pada Gambar 1.



Gambar 1 Alur Penelitian

### 2.1. Studi Literatur

Tahapan ini dilakukan studi literatur untuk mengumpulkan informasi yang relevan sesuai dengan topik penelitian ini. Pada tahapan ini melibatkan sumber-sumber terpercaya seperti jurnal ilmiah, buku, dan publikasi resmi. Tahapan ini dilakukan untuk mengetahui perkembangan dalam bidang penelitian yang bersangkutan serta memperoleh landasan teoritis yang kuat untuk membangun kerangka konseptual. Dalam konteks penelitian ini, studi literatur akan difokuskan pada klasifikasi aksara serta metode yang akan digunakan, yaitu *Histogram of Oriented Gradients* dan K-Nearest Neighbors (K-NN) sebagai teknik klasifikasi yang akan diterapkan. Oleh karena itu, tahap ini menjadi dasar penting dalam proses penelitian untuk memastikan bahwa metode yang dipilih dapat diterapkan dengan benar dan menghasilkan data yang valid.

### 2.2. Pengumpulan Data

Pada tahapan ini dilakukan pengumpulan data, data tersebut diambil dari website open source kaggle (url: <https://www.kaggle.com/datasets/asrulsaid/aksara-lontara>). Dataset tersebut disediakan oleh Asrul Said.

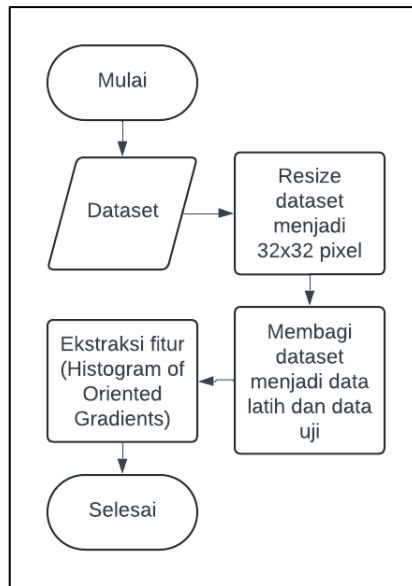
pada penelitian ini menggunakan dataset tersebut dan menggunakan 7542 citra Aksara Lontara. Tabel 1 merupakan persebaran kelas dan jumlah dataset.

Tabel 1 Persebaran Dataset

No	Kelas	Jumlah Data
1	A	324
2	Ba	324
3	Ca	324
4	Da	324
5	Ga	324
6	Ha	324
7	Ja	324
8	Ka	324
9	La	324
10	Ma	324
11	Mpa	324
12	Na	324
13	Nca	324
14	Nga	324
15	Mgla	324
16	Nra	324
17	Nya	324
18	Pa	324
19	Ra	324
20	Sa	324
21	Ta	324
22	Wa	324
23	Ya	324
<b>Jumlah</b>		7452

### 2.3. Pengolahan Data

Tahap berikutnya yang akan dilakukan pada penelitian ini yakni pengolahan data. Pada tahapan ini akan dilakukan preprocessing, splitting dataset, dan ekstraksi fitur. Pada tahapan ini perlu dilakukan preprocessing untuk membersihkan dan memperbaiki data seperti mengubah ukuran, memperbaiki data noise dan lainnya. Setelah preprocessing selesai data tersebut akan dibagi menjadi data latih dan data uji. Kemudian data latih dan data uji tersebut akan dilakukan ekstraksi fitur, dimana data mentah akan diproses menjadi fitur yang dapat digunakan untuk pemodelan atau klasifikasi. Pada penelitian ini akan menggunakan Histogram of Oriented Gradients sebagai ekstraksi fiturnya. Pada Gambar 2 merupakan tahapan pengolahan data pada penelitian ini.



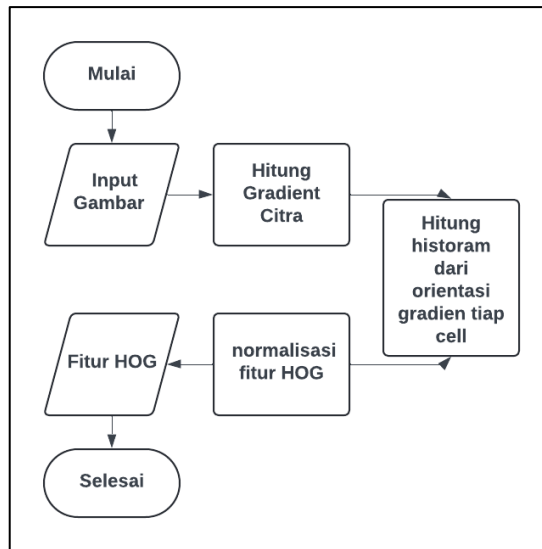
Gambar 2 Alur Pengolahan Data

Pada Gambar 2 merupakan alur pengolahan data yang dilakukan pada penelitian ini. Berikut penjelasan mengenai alur penelitian tersebut.

1. *Resize Dataset*, dalam tahapan ini ukuran awal pada dataset yang digunakan yakni  $128 \times 128$  pixel, dataset tersebut akan diubah ukurannya menjadi  $32 \times 32$  pixel. Tujuannya adalah untuk meningkatkan efisiensi komputasi dengan mengurangi ukuran data yang diperlukan, memastikan konsistensi ukuran data agar sesuai dengan kebutuhan model, mengurangi dimensi fitur untuk mengurangi kompleksitas model dan menghindari overfitting, serta meningkatkan kecepatan dan efisiensi dengan memperkecil ukuran data untuk memperpendek waktu yang dibutuhkan.
2. *Splitting Dataset*, pada tahapan ini dataset akan dibagi menjadi data latih dan data uji dengan rasio 90:10, 90% untuk data latih dan 10% untuk data uji. Jumlah data latih sebanyak 6706 dan data uji sebanyak 746 dari total 7452 citra Aksara Lontara
3. *Ekstraksi fitur*, pada tahapan ini menggunakan ekstraksi fitur *Histogram of Oriented Gradients*. *Histogram of Oriented Gradient*. Metode ini bekerja dengan membagi gambar menjadi sel sel kecil yang disebut sel sel *pixel* yang kemudian akan dihitung gradien orientasi dari setiap *pixel* dalam sel tersebut. Selanjutnya histogram orientasi dari gradien ini dihitung untuk masing masing sel. Hasil dari proses ini adalah deskripsi yang kompak tetapi kuat dari struktur dan pola yang ada dalam gambar

### 2.3.1. Histogram of Oriented Gradient

*Histogram of Oriented Gradients* merupakan descriptor fitur seperti deteksi tepi sobel, Canny, *Scale Invariant and Feature Transform*. *Histogram of Oriented Gradients* seringkali digunakan dalam bidang visi komputer dan memproses gambar sebagai deteksi objek. Metode ini mengukur frekuensi munculnya orientasi radian pada area gambar yang telah dilokalisasi. Pada Gambar 3 merupakan alur Histogram of Oriented Gradients.



Gambar 3 Alur Histogram of Oriented Gradients

Dari gambar proses pertama yang dilakukan adalah menghitung gradient citra dengan persamaan.

$$|G| = \sqrt{G_x^2 + G_y^2} \dots\dots\dots (1)$$

Keterangan :

G = Citra *Grayscale*

$G_x$  = Matrik terhadap sumbu x

$G_y$  = Matrik terhadap sumbu y

$G_x$  dan  $G_y$  dapat dihitung dengan cara

Setelah menghitung gradient citra Langkah selanjutnya adalah mentransformasikan gradient kedalam koordinat sumbu antara sumbu  $0^\circ$  hingga  $180^\circ$  yang disebut dengan orientasi gradient. Persamaan 2 merupakan persamaan orientasi gradient.

$$\theta = \arctan\left(\frac{G_x}{G_y}\right) \dots\dots\dots (2)$$

Setelah menghitung Orientasi Gradient pada setiap sel. Setiap *pixel* memiliki nilai histogramnya masing-masing berdasarkan nilai yang dihasilkan dalam memperhitungkan gradient yang akan dinormalisasi setiap bloknya. Kemudian nilai normalisasi fitur blok didapatkan dari rumus persamaan 3. Fitur blok yang ternormalisasi berguna untuk mengurangi efek perubahan kecerahan pada suatu blok.

$$b_n = \frac{b}{\sqrt{b^2+e}} \dots\dots\dots (3)$$

Keterangan :

$b_n$  = Nilai normalisasi fitur blok

b = Nilai fitur blok

e = Bilangan bulat positif

setelah menghitung normalisasi , langkah selanjutnya adalah menghitung setiap blok nilai yang telah ternormalisasi dan digabungkan menjadi satu vektor kemudian menghasilkan vektor *Histogram of Oriented Gradients*. Persamaan 4 merupakan digunakan untuk mendapatkan nilai fitur.

$$h_n = \frac{h}{\|h\|^2+e} \dots\dots\dots (4)$$

Keterangan :

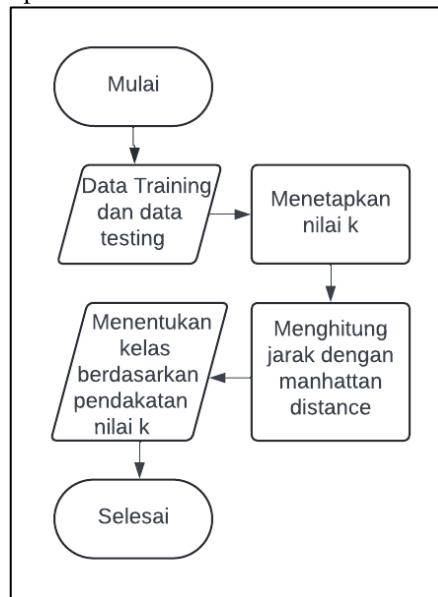
h = Nilai fitur

e = Bilangan bulat positif

#### 2.4. Model Klasifikasi

Setelah melalui tahapan pengolahan data seperti preprocessing, splitting dataset, dan ekstraksi fitur, maka akan dilanjutkan pembuatan model klasifikasi. Metode klasifikasi yang digunakan yakni K-NN. K-NN adalah algoritma pembelajaran yang dapat digunakan untuk mengklasifikasikan data. Cara kerja algoritma ini melibatkan perbandingan antara set data latih (*training set*) dengan set data uji (*testing set*) untuk mengidentifikasi jarak terdekat antara titik-titik data. Dalam menentukan kelas dari suatu titik data, algoritma ini menggunakan variabel

k dimana k merupakan jumlah tetangga terdekat yang akan dibandingkan. Biasanya, nilai k yang dipilih adalah bilangan ganjil. Pada Gambar 4 merupakan alur metode K-NN.



Gambar 4 Alur Metode K-NN

Pada Gambar 4 merupakan alur dari metode K-NN. Dalam penggunaan K-NN untuk klasifikasi adalah menetapkan nilai parameter k, nilai k merupakan jumlah tetangga terdekat yang akan digunakan untuk melakukan prediksi. Setelah menentukan nilai k, algoritma akan mencari tetangga terdekat dari data yang baru dengan memperhitungkan jarak antara data baru tersebut. Setelah itu menentukan kelas berdasarkan nilai k. Metrik jarak yang digunakan dalam penelitian ini adalah Manhattan Distance. Persamaan 6 merupakan rumus Manhattan Distance

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \dots\dots\dots (6)$$

Keterangan :

- d = jarak antara koordinat x dan y
- x, y = vektor x dan vektor y
- i = setiap data n = jumlah keseluruhan data
- $x_i, y_i$  = vektor x ke i dan vektor y ke i

### 2.5. Pengujian Model

Pada tahapan pengujian menggunakan confusion matrix. Confusion Matrix merupakan alat yang digunakan untuk mengevaluasi suatu model klasifikasi. Confusion Matrix menggambarkan jumlah prediksi yang benar (*True Positive* dan *True Negative*) serta jumlah prediksi yang salah (*False Positive* dan *False Negative*). Pada confusion matrix terdapat 4 istilah yakni, *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN). Pada Tabel 2 merupakan penjelasan mengenai confusion matrix.

Tabel 2 Confusion Matrix

Confusion Matrix		Nilai Aktual	
		Positif	Negatif
Nilai	Positif	TP	FP
	Negatif	FN	TN

Berdasarkan Tabel 2 dapat diketahui bahwa terdapat 4 istilah dalam confusion matriks. Berikut penjelasannya.

1. TP adalah data yang sebenarnya positif dan diprediksi sebagai positif.
2. TN adalah data yang sebenarnya negatif dan diprediksi sebagai negatif.
3. FP adalah data yang sebenarnya negatif namun diprediksi sebagai positif.
4. FN adalah data yang sebenarnya positif namun diprediksi sebagai negatif.

Dari keempat bagian tersebut akan diolah untuk mendapatkan metrik evaluasi yang berguna untuk mengukur tingkat performa dari sebuah algoritma. Berikut merupakan metrik tersebut.

1. *Accuray*

*Accuray* adalah metrik evaluasi yang berguna untuk menilai seberapa efektif model dalam melakukan prediksi yang tepat dari jumlah keseluruhan prediksi yang dilakukan. Pada Persamaan 7 digunakan untuk menghitung yang digunakan untuk menghitung *accuracy*.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \dots\dots\dots (7)$$

2. *Precision*

*Precision* adalah ukuran evaluasi yang memperhitungkan seberapa efisien model dalam melakukan prediksi yang tepat untuk kelas positif dibandingkan dengan keseluruhan prediksi positif yang dihasilkan. Pada Persamaan 8 digunakan untuk menghitung nilai *precision*.

$$Precision = \frac{TP}{(TP+FP)} \dots\dots\dots (8)$$

3. *Recall*

*Recall* adalah sebuah ukuran evaluasi yang digunakan untuk menunjukkan seberapa efektifnya suatu model dalam mengenali kelas positif secara tepat. Pada Persamaan 9 digunakan untuk menghitung nilai *Recall*.

$$Recall = \frac{TP}{(TP+FN)} \dots\dots\dots (9)$$

4. *F1-Score*

*F1-score* adalah sebuah metrik evaluasi yang menggabungkan antara presisi (*precision*) dan *recall* dalam sebuah formula tunggal. Pada Persamaan 10 digunakan untuk menghitung Nilai *F1-Score*.

$$F - 1 \text{ Score} = \frac{(2 \times Recall \times Precision)}{(Recall+Precision)} \dots\dots\dots (10)$$

3. HASIL DAN PEMBAHASAN

Pengujian yang telah dilakukan menggunakan ekstraksi fitur *Histogram of Oriented Gradients* dan klasifikasi K-NN dengan nilai k=3 menghasilkan sejumlah metrik evaluasi yang penting. Metrik-metrik ini memberikan gambaran yang jelas tentang seberapa baik model ini dapat memprediksi kelas dari data yang diberikan.

Tabel 3 Hasil Pengujian K-NN

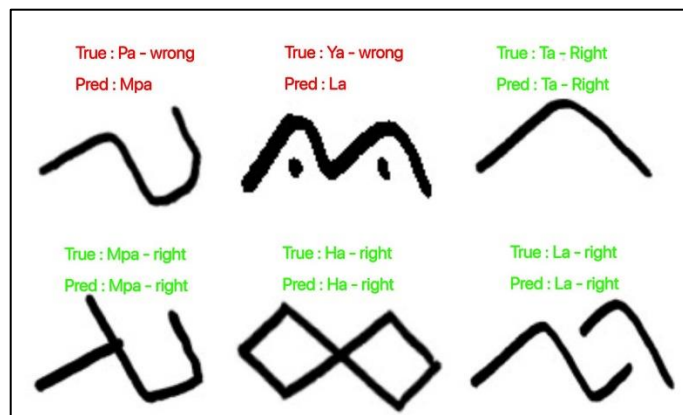
No	K-NN	
	Matrix	Nilai
1	Akurasi	0,9235
2	Presisi	0,9237
3	Recall	0,9227
4	F1-Score	0,9221

Dapat dilihat nilai akurasi (*accuracy*) dari model ini sebesar 0.9235 dan mendapatkan total data error sebanyak 110. Akurasi mengukur seberapa tepat model dalam mengklasifikasikan data secara keseluruhan. Dalam konteks pengujian ini, nilai akurasi yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam memprediksi kelas dari sampel data yang diberikan. Selanjutnya, kita memiliki nilai presisi (*precision*) sebesar 0.9237. Presisi mengukur seberapa banyak dari prediksi positif yang sebenarnya benar-benar positif. Dengan nilai presisi yang tinggi, ini menunjukkan bahwa model cenderung memberikan sedikit false positive, yaitu prediksi yang salah tentang kelas positif. Selain itu, terdapat nilai recall sebesar 0.9227. *Recall*, atau juga dikenal sebagai *sensitivity*, mengukur seberapa banyak dari kelas positif yang sebenarnya berhasil diprediksi oleh model. Dengan nilai recall yang tinggi, model mampu mengidentifikasi sebagian besar kelas positif yang ada dalam data. F1 Score, yang merupakan harmonic mean dari presisi dan recall, juga memberikan gambaran keseluruhan tentang performa model. Dalam kasus ini, *F1-Score* sebesar 0.9221 menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall, sehingga dapat dianggap sebagai model yang handal dalam memprediksi kelas dari data yang diberikan.

Evaluasi dengan nilai K = 3  
 Accuracy: 0.9235924932975871  
 Classification Report:

	precision	recall	f1-score	support
0	0.73	0.85	0.79	26
1	1.00	0.97	0.98	29
2	0.89	0.94	0.91	33
3	0.91	0.86	0.89	36
4	0.81	0.71	0.76	24
5	1.00	1.00	1.00	33
6	0.96	0.96	0.96	26
7	1.00	1.00	1.00	31
8	1.00	0.90	0.95	40
9	0.89	0.93	0.91	42
10	1.00	1.00	1.00	39
11	0.90	0.93	0.92	30
12	1.00	1.00	1.00	32
13	0.96	1.00	0.98	25
14	1.00	0.94	0.97	31
15	0.80	0.88	0.84	32
16	0.92	0.94	0.93	36
17	0.84	0.79	0.82	39
18	1.00	1.00	1.00	30
19	1.00	1.00	1.00	38
20	0.89	0.92	0.91	26
21	0.82	0.91	0.86	34
22	0.93	0.79	0.86	34
accuracy			0.92	746
macro avg	0.92	0.92	0.92	746
weighted avg	0.93	0.92	0.92	746

Gambar 5 Hasil Pengujian K-NN



Gambar 6 Citra Hasil Uji dan Prediksi

Pada Gambar 5 merupakan hasil dari pengujian K-NN. Hasil ini menunjukkan bahwa kombinasi antara ekstraksi fitur *Histogram of Oriented Gradients* dan klasifikasi K-NN dengan  $k=3$  dapat menghasilkan model yang cukup efektif dalam mengklasifikasikan data. Pada Gambar 6 merupakan contoh dari hasil klasifikasi citra yang error dan citra yang benar. Pada data yang error teks berwarna merah, sedangkan yang benar berwarna hijau. Meskipun begitu, perlu diingat bahwa performa model juga dapat dipengaruhi oleh berbagai faktor lain seperti ukuran dataset, distribusi kelas, dan proses preprocessing data. Oleh karena itu, penting dilakukannya evaluasi dan pengoptimalan model guna meningkatkan performa dan kehandalan prediksi.



#### 4. SIMPULAN

Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma K-NN dapat digunakan untuk mengenali pola Aksara Lontara dari dataset. Tidak hanya itu *Histogram of Oriented Gradients* juga dapat digunakan bersamaan dengan algoritma K-NN. Sebelum dilakukannya klasifikasi terdapat proses *preprocessing* dengan mengubah ukuran citra menjadi  $32 \times 32$  pixel. Setelah ukuran citra diubah dilakukan *splitting* dataset 90:10. Kemudian dilakukan ekstraksi fitur untuk mengekstrak fitur-fitur yang ada pada citra menggunakan *Histogram of Oriented Gradients*. Pada tahap *Histogram of Oriented Gradients* menggunakan ukuran sel  $8 \times 8$  pixel dan histogram bin berukuran 9. Hasil penelitian menunjukkan bahwa metode yang diterapkan mampu mendapatkan akurasi yang cukup tinggi yakni 0,9235.

#### 5. SARAN

Klasifikasi citra yang menggunakan K-NN dan ekstraksi fitur dapat dipengaruhi oleh ketepatan *preprocessing* seperti mengubah ukuran citranya agar dapat meningkatkan hasil akurasi, serta ekstraksi fitur yang digunakan. Semakin banyak variasi data input membutuhkan tahapan yang lebih tepat lagi, maka dari itu, masih terdapat peluang untuk penelitian lebih lanjut dalam menerapkan algoritma K-NN ini seperti menggunakan nilai k yang lainnya. Penelitian ini juga masih dapat dikembangkan lagi dengan menambahkan metode lainnya yang belum digunakan untuk mendapatkan akurasi yang lebih tinggi lagi.

#### DAFTAR PUSTAKA

- [1] A. F. Rabda, N. Suciati and D. A. Navastara, "Implementasi Metode Kombinasi Histogram of Oriented Gradients dan Hierarchical Centroid untuk Sketch Based Image Retrieval," *JURNAL TEKNIK ITS*, vol. 5, no. No. 2, pp. A311-A316, 2016.
- [2] A. Mulyanto, E. Susanti, F. Rosi, W. and R. I. Borman, "Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR)," *JEPIN*, vol. 7, no. No. 1, pp. 52-57, 2021.
- [3] A. P. Sari, H. Suzuki, T. Kitajima, T. Yasuno, D. A. Prasetya and Nachrowie, "Prediction Model of Wind Speed and Direction using Convolutional Neural Network - Long Short Term Memory," *IEEE International Conference on Power and Energy (PECon)*, pp. 356-361, 2020.
- [4] A. R. Widiarti and H. Suparwito, "Analisis Unjuk Kerja K-Nearest Neighbour untuk Klasifikasi Citra Aksara Bali Tulis Tangan," *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, vol. 12, no. No.2, pp. 123-132, 2022.
- [5] A. R. Widiarti, "Unjuk kerja k-nearest neighbor untuk alihaksara citra aksara Nusantara," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. No. 2, pp. 150-156, 2020.
- [6] F. Bimantoro, A. Aranta, G. S. Nugraha, R. Dwiyanaputra and A. Y. Husodo, "Pengenalan Pola Tulisan Tangan Aksara Bimamenggunakan CiriTekstur dan KNN," *J-COSINE*, vol. 5, no. No. 1, pp. 60-67, 2021.
- [7] F. Syuhada, "KLASIFIKASI CITRA TULISAN TANGAN AKSARA SASAK DENGAN METODE HISTOGRAM OF ORIENTED GRADIENTS DAN MULTINOMIAL LOGISTIC REGRESSION," *JITET*, vol. 11, no. No. 3, pp. 604-613, 2023.
- [8] F. T. Anggraeny, Y. V. Via and R. Mumpuni, "Image preprocessing analysis in handwritten Javanese," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. No. 2, pp. 860-867, 2023.
- [9] Fitrawahyudi and Kasmawati, "Kemampuan Bahasa Daerah Usia 17-22 Tahun:Proyeksi Kepunahan Bahasa Daerah di Kabupaten Maros," *IDIOMATIK*, vol. 1, no. No. 1, pp. 75-82, 2018.

- [10] I. I. Saputri, Purnawansyah and Herman, "Implementasi Metode Naïve Bayes Pada Pengenalan Tulisan Tangan Lontara," *Buletin Sistem Informasi dan Teknologi Islam*, vol. 2, no. No. 3, pp. 167-175, 2021.
- [11] I. P. R. Anandya, I. G. A. G. A. Kadyanan, I. D. M. B. A. Darmawan, C. R. A. Pramarta, A. A. I. N. E. Karyawati and N. A. S. ER, "Klasifikasi Aksara Bali Berbasis Suara Dengan Metode KNN dan FastDTW," *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 11, no. No. 3, pp. 581-586, 2023.
- [12] M. Tyagi, "HOG (Histogram of Oriented Gradients): An Overview," 4 Juli 2021. [Online]. Available: <https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f>.
- [13] N. M. Putry and B. N. Sari, "KOMPARASI ALGORITMA KNN DAN NAÏVE BAYES UNTUK KLASIFIKASI DIAGNOSIS PENYAKIT DIABETES MELITUS," *Evolusi: Jurnal Sains dan Manajemen*, vol. 10, no. No. 1, pp. 45-57, 2022.
- [14] P. Ginting, H. Rumapea, A. P. Silalahi, P. Lumbanraja and M. Aritonang, "Pengenalan Pola Aksara Karo Berdasarkan Citra Pola Menggunakan Metode K-Nearest Neighbor," *Methodika : Jurnal Ilmiah Teknik Informatika*, vol. 2, no. No. 1, pp. 38-45, 2022.
- [15] R. K. Wati and H. Irsyad, "Pengenalan Aksara Arab Menggunakan Metode JST Dengan Fitur HOG Dan LBP," *Jurnal Algoritme*, vol. 2, no. No. 1, pp. 39-54, 2021.
- [16] R. Yulianti, G. P. S. Wijaya and F. Bimantoro, "Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Moment Invariant dan Support Vector Machine," *J-COSINE*, vol. 3, no. No. 2, pp. 91-98, 2019.
- [17] S. Alfarisi and Subandi, "Implementasi Pengenalan Aksara Bali Menggunakan Direction Feature Extraction dan K-Nearest Neighbor," *Jurnal TICOM*, vol. 11, no. No. 1, pp. 50-54, 2022.
- [18] V. Oktavia and N. Wijaya, "Pengenalan Tulisan Tangan Huruf Latin Bersambung Menggunakan Local Binary Pattern dan K-Nearest Neighbor," *Jurnal Informatika Sunan Kalijaga*, vol. 7, no. No. 3, pp. 211-225, 2022.
- [19] Y. Sugianela and N. Suciati, "Ekstraksi fitur pada pengenalan karakter Aksara Jawa berbasis Histogram of Oriented Gradient," *Jurnal Ilmiah Teknologi Informasi*, vol. 17, no. No. 1, pp. 64-72, 2019.