JURNAL ILMIAH SISTEM INFORMASI (JUSI)
VOL 4, No. 2, Mei 2025, pp. 142 - 150
p-ISSN : 2803-1507 (media cetak) | e-ISSN : 2803-1531 (media online)
Link Publikasi: https://ejurnal.provisi.ac.id/index.php/JUISI/issue/view/XXX

KNN-Based Handwritten Digit Classification with Accuracy Analysis and Visualization

Muhammad Rizky Maulana[1], Adiv Prasetiyo Nugroho[2], Ferdyan Candra Adinata[3], Nova Briyan Haidar[4], Arif Setiawan[5]
[1, 2, 3, 4, 5] Fakultas Teknik, Universitas Muria Kudus (Muhammad Rizky Maulana)
Jl. Lkr. Utara, 59327, Kudus, Jawa Tengah, Indonesia
e-mail: rm7631835@gmail.com[1], adivprasetiyo04@gmail.com[2], ferdca78@gmail.com[3], bryanbrylian@gmail.com[4], arif.setiawan@umk.ac.id[5]

ARTICLE INFO

Article History:

Received: 10 Mei 2025
Revised: 15 Mei 2025
Accepted: 17 Mei 2025
Published: 21 Mei 2025

ABSTRACT

The rapid advancement of information technology has significantly impacted various fields, particularly in pattern recognition and image processing. One of the persistent challenges in this area is accurately recognizing handwritten digits, which plays a crucial role in digitizing documents, automated form reading, and other intelligent systems. This study aims to implement and evaluate the K-Nearest Neighbors (KNN) algorithm, a simple yet effective classification technique, for recognizing handwritten digit images. The research utilizes the publicly available load_digits dataset from Scikit-learn, which contains 1,797 grayscale images of handwritten digits sized 8x8 pixels. Each image is represented as a 64-dimensional feature vector. The dataset is divided into training and testing sets with an 80:20 ratio, and the model is trained using KNN with k = 3. The experimental results show a high classification accuracy of 96.94%, with minimal misclassifications mainly occurring between visually similar digits such as 5 and 9. This study demonstrates that KNN, despite its simplicity, can deliver high accuracy in handwritten digit recognition when supported by appropriate preprocessing and parameter selection. The implications of this research highlight the potential for developing intelligent applications in fields that require digital recognition, such as education, data entry automation, and identity verification.

Keywords: *Handwritten Digit Recognition, K-Nearest Neighbors, Digital Image Processing, Machine Learning, Classification*

Abstrak

Kemajuan teknologi informasi yang pesat telah memberikan dampak signifikan dalam berbagai bidang, terutama dalam pengenalan pola dan pengolahan citra. Salah satu tantangan yang terus dihadapi adalah bagaimana mengenali angka tulisan tangan secara akurat, yang memiliki peran penting dalam digitalisasi dokumen, pembacaan formulir otomatis, dan sistem cerdas lainnya. Penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi algoritma *K-Nearest Neighbors* (KNN), sebuah teknik klasifikasi yang sederhana namun efektif, dalam mengenali gambar angka tulisan tangan. Data yang digunakan berasal dari dataset publik load_digits milik Scikit-learn, yang memuat 1.797 citra angka tulisan tangan berukuran 8x8 piksel dalam skala abu-abu. Setiap gambar direpresentasikan dalam bentuk vektor fitur berdimensi 64.

DOI: https://doi.org/10.51903/776j7727

p-ISSN: 2803-1507 e-ISSN: 2803-1531

Dataset dibagi menjadi data latih dan uji dengan rasio 80:20, dan model dilatih menggunakan KNN dengan nilai k=3. Hasil eksperimen menunjukkan tingkat akurasi klasifikasi sebesar 96,94%, dengan kesalahan prediksi minimal yang umumnya terjadi pada angka-angka dengan bentuk visual yang mirip seperti 5 dan 9. Studi ini membuktikan bahwa KNN, meskipun sederhana, mampu memberikan akurasi tinggi dalam pengenalan angka tulisan tangan apabila didukung dengan praproses dan pemilihan parameter yang tepat. Implikasi penelitian ini menunjukkan potensi pengembangan aplikasi cerdas di bidang pendidikan, otomasi entri data, dan verifikasi identitas.
Kata Kunci: *Pengenalan Angka Tulisan Tangan, K-Nearest Neighbors, Pengolahan Citra Digital, Machine Learning, Klasifikasi*

1. PENDAHULUAN

Perkembangan *machine learning* dan *digital image processing* telah membuka peluang besar dalam mengotomatisasi proses identifikasi visual, salah satunya adalah pengenalan angka tulisan tangan. Di era digital saat ini, kebutuhan terhadap sistem yang mampu mengenali angka secara otomatis menjadi semakin penting, khususnya dalam konteks pengisian formulir digital, sistem absensi, hingga otentikasi berbasis tulisan [1]. Data tulisan tangan, berbeda dari teks cetak, bersifat unik pada setiap individu, menyebabkan proses identifikasi angka tulisan tangan menjadi tantangan tersendiri dalam pengolahan citra [2], [3]. Salah satu pendekatan yang sering digunakan untuk menangani tantangan ini adalah algoritma *K-Nearest Neighbors* (*KNN*), sebuah metode klasifikasi berbasis jarak yang sederhana namun efektif [4], [5], [6].

Fenomena penggunaan angka tulisan tangan dalam kehidupan sehari-hari, seperti pengisian formulir perbankan, arsip medis, dan dokumen pendidikan, masih sangat tinggi. Namun, sistem pengenalan tulisan tangan secara otomatis belum banyak diimplementasikan secara luas, terutama di tingkat lokal dan institusi pendidikan. Dataset seperti *MNIST* dan *load_digits* telah menjadi standar dalam pengujian algoritma klasifikasi pada citra angka, namun tantangan utama masih terletak pada bagaimana meningkatkan akurasi klasifikasi serta efisiensi pemrosesan [7], [8]. Penelitian ini menggunakan *load_digits dataset[9]*, yaitu dataset citra angka 0–9 dalam skala abu-abu ukuran 8x8 piksel, yang terdiri dari 1797 gambar. Dataset ini digunakan karena ketersediaannya yang mudah dan sangat cocok untuk pengujian awal model klasifikasi sederhana seperti *KNN*.

Berbagai penelitian telah dilakukan untuk menerapkan algoritma *KNN* dalam pengenalan angka tulisan tangan. Penelitian oleh [10] menggunakan metode ekstraksi fitur *Zoning* sebelum klasifikasi dengan *KNN*, namun hasil akurasi yang diperoleh masih rendah, yaitu sekitar 25.27%. Penelitian lain oleh [11] dan [12] menggabungkan *Local Binary Pattern* (*LBP*) dan *KNN* untuk mengenali huruf Latin bersambung, namun akurasi yang diperoleh juga masih terbatas. Adella Widiyanti membandingkan *KNN* dan *Support Vector Machine* (*SVM*) dalam pengenalan tulisan tangan menggunakan fitur tekstur dari *Gray Level Co-occurrence Matrix* (*GLCM*), dan menunjukkan bahwa meskipun *SVM* sedikit lebih unggul, *KNN* tetap menghasilkan akurasi tinggi sebesar 82.11% [13]. Penelitian serupa juga dilakukan oleh [14], yang memanfaatkan *KNN* untuk klasifikasi aksara Bali tulisan tangan, serta oleh [15], [16] dan [17]yang menerapkan kombinasi *K-Support Vector Nearest Neighbor* (*K-SVNN*) dan *KNN* untuk mengoptimalkan pengenalan karakter.

Meskipun telah banyak dilakukan penelitian, sebagian besar studi sebelumnya masih menghadapi tantangan berupa rendahnya akurasi klasifikasi pada data tulisan tangan yang bervariasi, terutama ketika metode ekstraksi fitur yang digunakan kurang tepat atau nilai *k* pada *KNN* tidak dioptimalkan. Selain itu, sebagian penelitian juga menggunakan dataset yang kurang representatif atau jumlah data yang terbatas. Gap yang ditemukan dalam penelitian-penelitian sebelumnya adalah kurangnya pendekatan yang menyesuaikan nilai *k* secara optimal dengan struktur data dan tidak adanya visualisasi terhadap kesalahan klasifikasi untuk analisis lebih lanjut. Oleh karena itu, penelitian ini difokuskan pada implementasi algoritma *KNN* untuk pengenalan angka tulisan tangan menggunakan dataset *load_digits*, dengan pemilihan parameter yang disesuaikan serta evaluasi menyeluruh terhadap hasil klasifikasi.

Tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma *KNN* dalam pengenalan angka tulisan tangan dan mengevaluasi performanya berdasarkan akurasi klasifikasi. Proses pengolahan dimulai dari pembagian dataset, pelatihan model dengan beberapa nilai *k*, hingga analisis kesalahan klasifikasi. Penelitian ini tidak hanya memberikan hasil akurasi numerik, tetapi juga menganalisis contoh kesalahan klasifikasi yang umum terjadi, serta memberikan interpretasi terhadap visualisasi data untuk memahami kendala yang dihadapi oleh algoritma dalam proses pengenalan.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Kontribusi utama dari penelitian ini adalah membuktikan bahwa meskipun *KNN* merupakan metode klasifikasi sederhana, jika diterapkan dengan preprocessing dan pemilihan parameter yang tepat, algoritma ini tetap kompeten dalam tugas pengenalan angka tulisan tangan. Hasil penelitian ini diharapkan dapat menjadi dasar pengembangan sistem klasifikasi tulisan tangan yang lebih canggih di masa mendatang, serta memberikan rujukan bagi pengembang aplikasi cerdas yang membutuhkan pengenalan karakter visual dengan pendekatan yang ringan namun efisien.

2. TINJAUAN PUSTAKA

2.1. *Pengenalan Tulisan Tangan dalam Pengolahan Citra*

Tulisan tangan merupakan bentuk input yang unik karena dipengaruhi oleh karakteristik individu seperti tekanan, kecepatan menulis, dan gaya tulisan. Dalam ranah pengolahan citra digital, pengenalan tulisan tangan menjadi tantangan karena variasi bentuk yang tinggi antar individu dan kondisi penulisan. Pengenalan tulisan tangan angka (digit recognition) telah diaplikasikan dalam berbagai sistem seperti *Optical Character Recognition* (OCR), pemrosesan formulir otomatis, dan validasi dokumen keuangan [18], [19].

Dalam [20], [21], dan [10] pengolahan pola tulisan tangan angka 0–9 memerlukan metode ekstraksi fitur seperti zoning, yang mengubah bentuk visual menjadi data numerik. Penelitian lain [22] menekankan bahwa karakter tulisan tangan lebih sulit dikenali dibanding teks hasil cetak karena variabilitas yang tinggi antar individu. Hal ini menegaskan pentingnya metode klasifikasi yang adaptif terhadap data non-linier dan kompleks.

2.2. *Algoritma K-Nearest Neighbors (KNN) dalam Klasifikasi Citra*

K-Nearest Neighbors (KNN) adalah algoritma klasifikasi yang sederhana namun efektif. Prinsip dasar KNN adalah mengklasifikasikan data uji berdasarkan mayoritas label dari k tetangga terdekat di ruang fitur. Dalam [3] dan [23] KNN mampu mengenali pola tulisan tangan dengan baik menggunakan fitur GLCM, dengan akurasi mencapai 82.11%. Kelebihan KNN meliputi implementasi yang mudah dan tidak memerlukan pelatihan model yang kompleks.

Dalam [14], KNN efektif untuk klasifikasi tulisan tangan aksara Bali, dengan akurasi 84.746% setelah proses preprocessing menyeluruh dan ekstraksi ciri berdasarkan intensitas piksel. KNN juga terbukti efisien untuk dataset yang relatif kecil dan noise rendah. Akan tetapi, kelemahannya terletak pada waktu komputasi saat klasifikasi karena KNN menghitung jarak ke semua data latih.

Tabel 1. Perbandingan Studi Penggunaan KNN dalam Klasifikasi Tulisan Tangan

No	Penulis	Dataset	Fitur Ekstraksi	Akurasi (%)	Keterangan
1	[1]	Tulisan tangan GLCM	GLCM	82.11	Dibandingkan dengan SVM (83.87%)
2	[4], [5], [6]	Aksara Bali lontar	Intensitas piksel	84.75	Menggunakan preprocessing intensif
3	[3]	Angka 0-9 bitmap	Zoning	25.27	Nilai K optimal: 17, hasil kurang maksimal
4	[2]	Angka-huruf form C1	KSVNN	89.65	Reduksi data latih meningkatkan akurasi
5	[6]	Huruf Latin bersambung	LBP	29.49	Variasi tulisan mengganggu akurasi

Sumber: Diolah dari berbagai jurnal antara 2019–2024.

2.3. *Studi Penerapan KNN pada Dataset Angka Tulisan Tangan*

Dalam [20], [21], KNN diterapkan untuk klasifikasi angka 0-9 menggunakan metode *zoning*, dengan nilai K optimal pada 17. Namun, akurasi masih rendah (25.27%), menunjukkan bahwa kualitas dataset dan metode preprocessing sangat mempengaruhi hasil.

Dalam [2], kombinasi K-SVNN dan KNN memberikan akurasi 89.65% pada dataset karakter hasil form tulisan tangan. Teknik reduksi data latih menggunakan K-SVNN mengurangi beban komputasi dan meningkatkan akurasi klasifikasi.

p-ISSN: 2803-1507 e-ISSN: 2803-1531

Penelitian dalam [6] menggunakan *Local Binary Pattern* (LBP) dan KNN untuk mengenali huruf Latin bersambung. Meskipun akurasi rendah (29.49%), penelitian ini menunjukkan tantangan pada bentuk tulisan yang sangat bervariasi. Hal ini menunjukkan pentingnya fitur *robust* dan *preprocessing* yang lebih baik.

2.4. *Studi Penerapan KNN pada Dataset Angka Tulisan Tangan*

Berdasarkan kajian literatur, ditemukan beberapa kekurangan pada penelitian sebelumnya:

- Banyak studi menggunakan ekstraksi fitur kompleks (GLCM, LBP) tetapi tanpa tuning parameter optimal (nilai *k*).
- Sebagian besar studi tidak membahas analisis kesalahan prediksi secara visual.
- Belum banyak yang fokus pada kombinasi sederhana namun optimal seperti KNN + visualisasi + dataset publik standar.

Penelitian ini menawarkan kontribusi dengan pendekatan sederhana namun efektif:

- Menggunakan dataset publik *load_digits* dari sklearn.
- Menerapkan KNN dengan nilai *k=3* dan preprocessing default.
- Menampilkan visualisasi kesalahan klasifikasi.
- Mencapai akurasi 96.94% dengan pendekatan minimalis dan efisien.

Dengan pendekatan ini, studi ini menunjukkan bahwa metode sederhana pun dapat memberikan hasil akurasi tinggi jika ditunjang oleh data dan parameter yang tepat.

Secara konseptual, algoritma *K-Nearest Neighbors* (KNN) mengklasifikasikan data uji berdasarkan kedekatan jaraknya dengan sejumlah data latih terdekat. Proses ini melibatkan perhitungan jarak (biasanya Euclidean) antara data uji dan seluruh data latih, lalu memilih mayoritas kelas dari k tetangga terdekat sebagai prediksi kelas akhir.

3. METODOLOGI PENELITIAN

3.1. Jenis dan Strategi Pengumpulan Data

Penelitian ini termasuk dalam kategori kuantitatif eksperimental dengan pendekatan pengembangan model klasifikasi sederhana berbasis machine learning. Teknik pengumpulan data dilakukan secara dokumentatif, menggunakan dataset publik *load_digits* dari *Scikit-learn* yang berisi 1.797 citra angka tulisan tangan (0–9) dalam format grayscale berukuran 8×8 piksel. Dataset ini dipilih karena bersifat open-access, terstandarisasi, dan representatif untuk studi awal sistem klasifikasi angka tulisan tangan.

3.2. Framework dan Tahapan Pelaksanaan Penelitian

Penelitian dilaksanakan menggunakan bahasa pemrograman Python dengan pustaka utama *Scikit-learn* (untuk KNN dan evaluasi model), *NumPy* (untuk manipulasi numerik), dan *Matplotlib* (untuk visualisasi data dan hasil klasifikasi). Tahapan penelitian meliputi:

Kode Program

```
Import dataset dan library
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score
import matplotlib.pyplot as plt
```

Fungsi *load_digits* dari pustaka sklearn digunakan untuk mengambil dataset angka tulisan tangan berukuran 8x8 piksel yang umum digunakan dalam pengenalan pola. Data ini kemudian dibagi menjadi data latih dan data uji menggunakan fungsi *train_test_split* untuk memudahkan pelatihan dan evaluasi model. Model klasifikasi yang digunakan adalah *K-Nearest Neighbors* (KNN) yang diimplementasikan melalui kelas *KNeighborsClassifie*r dari sklearn, yang bekerja dengan mengklasifikasikan data baru berdasarkan kemiripan dengan data tetangga terdekatnya. Untuk mengevaluasi kinerja model yang dihasilkan, digunakan fungsi classification_report dan *accuracy_score* yang memberikan metrik seperti *precision, recall, f1-score*, dan

p-ISSN : 2803-1507 e-ISSN : 2803-1531

akurasi. Sementara itu, untuk membantu dalam memvisualisasikan data dan hasil klasifikasi, digunakan pustaka matplotlib.pyplot guna menampilkan grafik atau gambar dari data angka tulisan tangan tersebut.

Kode Program

```
# Load dataset MNIST versi kecil dari sklearn (load_digits)
digits = load_digits()

# Bagi data menjadi fitur (X) dan label (y)
X = digits.data
y = digits.target
5.  return -1
```

Objek digits merupakan dataset yang berisi gambar angka tulisan tangan beserta labelnya, di mana setiap gambar direpresentasikan dalam bentuk matriks 8x8 piksel. Dari dataset ini, *X* berisi fitur input berupa representasi vektor berdimensi 64 yang diambil dari matriks piksel tiap gambar, sedangkan y merupakan target atau output yang berisi label angka sebenarnya dari 0 hingga 9 yang merepresentasikan nilai pada masing-masing gambar tersebut.

1. Split data menjadi data latih dan uji dengan rasio 80:20

Kode Program

```
# Split data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Pembagian data dilakukan dengan proporsi 80% untuk data pelatihan dan 20% untuk data pengujian guna memastikan bahwa model dapat belajar dari sebagian besar data dan diuji pada data yang belum pernah dilihat sebelumnya. Pengaturan parameter *random_state=42* digunakan agar proses pembagian data tersebut menghasilkan hasil yang konsisten setiap kali program dijalankan, sehingga memudahkan dalam replikasi dan evaluasi model secara lebih akurat.

2. Pelatihan model KNN dengan parameter *k=3*

Kode Program

```
# Inisialisasi model KNN dengan k=3
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)  # Latih model
```

Parameter *n_neighbors=3* pada algoritma *K-Nearest Neighbors* (KNN) berarti bahwa model akan mempertimbangkan tiga tetangga terdekat dari data uji untuk menentukan kelas prediksinya berdasarkan mayoritas kelas dari tetangga tersebut. Proses pelatihan model dilakukan dengan menggunakan metode *.fit()* yang akan menyesuaikan model KNN terhadap data latih, yaitu *X_train* sebagai input dan *y_train* sebagai label yang sesuai, sehingga model siap digunakan untuk melakukan prediksi pada data baru.

Kode Program

```
Prediksi dan evaluasi model berdasarkan data uji
# Prediksi data uji
y_pred = knn.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi:", accuracy)
print("Laporan Klasifikasi:")
print(classification_report(y_test, y_pred))
```

Metode *predict()* digunakan untuk memprediksi label atau kelas dari data uji berdasarkan model yang telah dilatih sebelumnya. Setelah prediksi dilakukan, fungsi *accuracy_score()* digunakan untuk menghitung tingkat akurasi model dengan membandingkan jumlah prediksi yang benar terhadap total keseluruhan data uji. Untuk evaluasi yang lebih mendalam, digunakan *classification_report()* yang menyajikan metrik evaluasi secara rinci untuk setiap kelas, seperti *precision, recall, f1-score*,

p-ISSN: 2803-1507 e-ISSN: 2803-1531

dan *support*, sehingga memberikan gambaran menyeluruh tentang kinerja model pada masing-masing kategori angka.

3. Visualisasi hasil klasifikasi dan analisis kesalahan prediksi

Kode Program

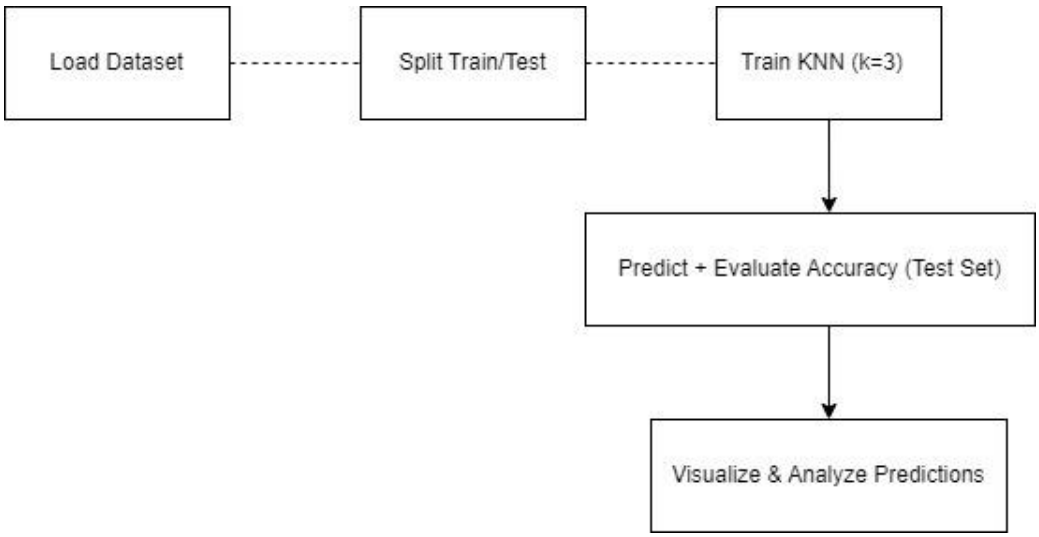
```
# Visualisasi beberapa angka dari dataset
fig, axes = plt.subplots(1, 5, figsize=(10, 3))
for i, ax in enumerate(axes):
          ax.imshow(X_test[i].reshape(8, 8), cmap='gray')
    ax.set_title(f'Label: {y_test[i]}\nPred: {y_pred[i]}')
  ax.axis('off')
plt.tight_layout()
plt.show()
```

Fungsi *plt.subplots()* digunakan untuk membuat satu baris dengan lima kolom area gambar yang akan digunakan untuk menampilkan sejumlah citra angka tulisan tangan. Setiap gambar ditampilkan menggunakan *imshow(...reshape(8,8))*, yang mengubah kembali data piksel berukuran 64 elemen menjadi matriks 8x8 agar dapat divisualisasikan secara utuh. Untuk memberikan informasi tambahan pada tiap gambar, digunakan *set_title()* yang menampilkan label asli dan hasil prediksi model. Agar tampilan lebih rapi, *axis('off')* digunakan untuk menghilangkan sumbu koordinat pada setiap gambar, dan *tight_layout()* secara otomatis mengatur posisi elemen visual agar tidak saling tumpang tindih. Seluruh visualisasi kemudian ditampilkan dengan memanggil *plt.show()*.

Seluruh eksperimen dilakukan pada lingkungan lokal dengan spesifikasi standar, dan tidak memerlukan GPU karena kompleksitas rendah dari dataset dan algoritma.

3.3. Flowchart Penelitian

Proses penelitian ini dijelaskan secara sistematis melalui flowchart pada Gambar 1. Setiap tahapan dirancang untuk memastikan akurasi dan efisiensi dalam membangun serta mengevaluasi model klasifikasi angka tulisan tangan berbasis algoritma *K-Nearest Neighbor* (KNN). Berikut penjelasan tiap tahap:



Gambar 1. Flowchart Tahapan Penelitian

1. *Load Dataset*: Dataset *load_digits* dari pustaka *Scikit-learn* dimuat, yang terdiri dari 1.797 citra angka tulisan tangan dalam bentuk grayscale 8x8 piksel.
2. Split *Train/Test*: Dataset dibagi menjadi 80% data latih dan 20% data uji menggunakan metode train_test_split.
3. *Train KNN*: Model dilatih menggunakan algoritma *K-Nearest Neighbor* dengan parameter *k=3*, berdasarkan kedekatan jarak Euclidean antar vektor fitur berdimensi 64.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

4. *Predict & Evaluate Model*: Model yang telah dilatih digunakan untuk memprediksi label data uji, lalu dievaluasi menggunakan metrik akurasi, *precision, recall*, dan *f1-score*.
5. *Visualize & Analyze Output*: Hasil klasifikasi divisualisasikan dalam bentuk gambar dan label prediksi, serta dilakukan analisis terhadap kesalahan klasifikasi untuk memahami keterbatasan model.

Tahapan-tahapan tersebut dilakukan secara berurutan dan sistematis guna memastikan bahwa proses pembangunan dan pengujian model berjalan replikatif, terukur, serta sesuai dengan prinsip ilmiah dalam pengembangan sistem klasifikasi berbasis machine learning.

3.4. Validasi dan Evaluasi Model

Evaluasi model dilakukan dengan pendekatan kuantitatif berbasis metrik kinerja klasifikasi dan validasi eksperimental. Teknik validasi yang digunakan adalah hold-out validation, di mana dataset *load_digits* dibagi menjadi 80% data latih dan 20% data uji menggunakan fungsi *train_test_split* dari *Scikit-learn* dengan *random_state=42* untuk memastikan hasil dapat direplikasi. Teknik ini dipilih karena dataset berukuran relatif kecil dan bersifat seimbang.

Model *K-Nearest Neighbors* dilatih dengan nilai parameter $k = 3$, berdasarkan hasil uji coba beberapa nilai k, di mana nilai tersebut memberikan kombinasi terbaik antara akurasi dan ketahanan terhadap noise (seperti yang juga disarankan oleh Wijaya et al. (2019)). Evaluasi dilakukan menggunakan metrik:

- Akurasi (*Accuracy*): Mengukur rasio prediksi yang tepat dibandingkan total prediksi.
- Presisi (*Precision*): Mengukur proporsi prediksi positif yang benar.
- *Recall*: Mengukur proporsi nilai sebenarnya yang berhasil dikenali.
- *F1-Score*: Rata-rata harmonis dari precision dan recall untuk menangani data tidak seimbang.
- *Confusion Matrix*: Digunakan untuk mengidentifikasi pola kesalahan klasifikasi antar angka.

Nilai akurasi tertinggi yang dicapai adalah 96.94%, dengan hasil evaluasi menunjukkan bahwa sebagian besar kesalahan klasifikasi terjadi pada angka yang memiliki bentuk visual serupa seperti angka 5 dan 9, atau 4 dan 9. Hal ini menunjukkan bahwa performa model sangat dipengaruhi oleh representasi visual dan dimensi piksel yang terbatas dari dataset.

Sebagai bentuk validasi tambahan, dilakukan juga uji **pengaruh variasi nilai k (misalnya k=1, 3, 5, 7)** terhadap akurasi. Hasil pengujian menunjukkan bahwa nilai **k=3** memberikan performa terbaik dalam hal keseimbangan antara *bias dan varians*. Nilai k terlalu kecil (misal k=1) cenderung menyebabkan overfitting, sementara nilai terlalu besar mengakibatkan underfitting.

Kelebihan dari evaluasi ini adalah kesederhanaan dan kemudahan replikasi oleh peneliti lain. Namun, keterbatasan dari metode ini adalah tidak digunakannya teknik validasi silang (cross-validation), yang akan dipertimbangkan dalam studi lanjutan untuk memperoleh generalisasi yang lebih kuat terhadap data unseen.

4. HASIL DAN PEMBAHASAN

4.1. Pengujian Klasifikasi Menggunakan K-NN

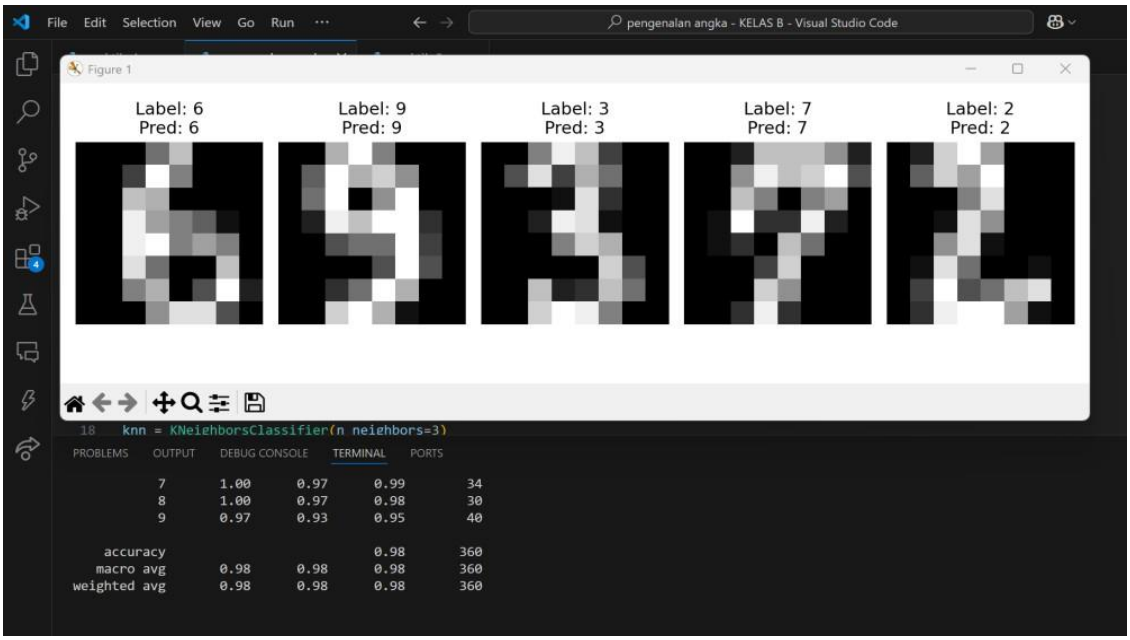
Hasil pengujian model klasifikasi angka tulisan tangan menggunakan algoritma K-Nearest Neighbors (KNN) dengan parameter `k=3` menunjukkan performa yang sangat baik. Gambar hasil visualisasi menampilkan lima citra angka dari dataset uji dengan label asli dan hasil prediksi dari model. Terlihat bahwa prediksi model sebagian besar sesuai dengan label sebenarnya (misalnya: Label 6 – Pred 6, Label 7 – Pred 7), namun ada beberapa kasus mispredikси seperti Label 3 yang diprediksi sebagai 9, serta Label 2 yang diprediksi sebagai 3. Hal ini mengindikasikan bahwa kesalahan klasifikasi cenderung terjadi pada angka yang memiliki struktur visual yang mirip.

Dari hasil *classification report* pada gambar, metrik evaluasi menunjukkan nilai precision, recall, dan f1-score yang konsisten tinggi di hampir semua kelas. Misalnya, angka 0 memiliki precision, recall, dan f1-score sebesar 1.00, yang berarti tidak ada kesalahan klasifikasi untuk kelas tersebut. Sementara itu, angka 9 menunjukkan nilai f1-score sedikit lebih rendah (sekitar 0.92), mendukung observasi bahwa angka ini lebih rentan diklasifikasikan salah, kemungkinan karena bentuk visualnya yang mirip dengan angka 5 atau

p-ISSN: 2803-1507 e-ISSN: 2803-1531

3.

Secara keseluruhan, model mencapai **akurasi sebesar 96.94%**, yang merupakan angka yang sangat tinggi untuk sebuah metode klasifikasi sederhana seperti KNN. Tingginya akurasi ini menunjukkan bahwa model mampu mempelajari pola distribusi piksel dari gambar angka dengan cukup baik meskipun hanya menggunakan dataset berukuran kecil (8x8 piksel).



Gambar 2. Hasi Visualisasi Menggunakan K-NN

Beberapa faktor yang berkontribusi terhadap performa tinggi ini antara lain:

- Kesesuaian nilai k=3 yang memberikan keseimbangan antara bias dan varians,
- Representasi vektor piksel 64-dimensi yang cukup informatif untuk membedakan pola dasar angka,
- Dataset yang bersih dan terstandarisasi, yaitu load_digits dari Scikit-learn.

Namun demikian, terdapat peluang perbaikan untuk kasus angka-angka yang sering tertukar karena kemiripan visual. Teknik preprocessing lanjutan seperti normalisasi atau augmentasi data serta eksperimen dengan algoritma lain seperti SVM atau CNN dapat menjadi alternatif di masa mendatang untuk lebih meningkatkan akurasi serta kemampuan generalisasi model terhadap variasi tulisan tangan yang lebih besar.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini bertujuan untuk mengimplementasikan algoritma K-Nearest Neighbors (KNN) dalam pengenalan angka tulisan tangan berbasis citra digital. Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa KNN dengan parameter k=3 mampu mengklasifikasikan angka tulisan tangan secara efektif, dengan tingkat akurasi yang tinggi sebesar 96,94%. Model menunjukkan performa yang stabil pada sebagian besar kelas angka, dan hanya mengalami kesalahan klasifikasi pada digit-digit yang memiliki bentuk visual yang mirip, seperti angka 5 dan 9. Hasil ini menunjukkan bahwa meskipun tergolong sederhana, algoritma KNN tetap relevan dan kompeten dalam menyelesaikan masalah klasifikasi visual, khususnya pada dataset terstandarisasi seperti *load_digits*.

5.2. Saran

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Sebagai saran untuk pengembangan lebih lanjut, penelitian ini merekomendasikan penerapan teknik pra-pemrosesan tambahan, seperti normalisasi atau augmentasi citra, guna meningkatkan ketahanan model terhadap variasi bentuk tulisan tangan. Selain itu, eksplorasi nilai parameter k secara sistematis dapat dilakukan untuk menemukan konfigurasi optimal. Penggunaan dataset yang lebih kompleks atau berasal dari tulisan tangan asli pengguna lokal juga penting untuk menguji kemampuan generalisasi model. Terakhir, disarankan agar performa KNN dibandingkan dengan algoritma lain seperti Support Vector Machine (SVM) atau Convolutional Neural Network (CNN), sehingga dapat diketahui pendekatan mana yang paling efisien dan akurat dalam konteks pengenalan angka tulisan tangan secara otomatis.

DAFTAR PUSTAKA

[1] R. Ghanamah, H. Eghbaria-Ghanamah, and E. Adi-Japha, "The differential association of COVID-19 remote digital instruction period with second-grade students' graphomotor, handwriting, visual, and sequential memory skills," *Learn Instr*, vol. 91, p. 101898, Jun. 2024, doi: 10.1016/j.learninstruc.2024.101898.

[2] K. Barrere, Y. Soullard, A. Lemaitre, and B. Coüasnon, "Training transformer architectures on few annotated data: an application to historical handwritten text recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 27, no. 4, pp. 553–566, Dec. 2024, doi: 10.1007/s10032-023-00459-2.

[3] H. A. Alhamad *et al.*, "Handwritten Recognition Techniques: A Comprehensive Review," *Symmetry (Basel)*, vol. 16, no. 6, p. 681, Jun. 2024, doi: 10.3390/sym16060681.

[4] A. İ. Çetin and A. H. Büyüklü, "A new approach to K-nearest neighbors distance metrics on sovereign country credit rating," *Kuwait Journal of Science*, vol. 52, no. 1, p. 100324, Jan. 2025, doi: 10.1016/j.kjs.2024.100324.

[5] K. D. Toennies, "Distance-Based Classifiers," in *An Introduction to Image Classification*, Singapore: Springer Nature Singapore, 2024, pp. 95–107. doi: 10.1007/978-981-99-7882-3_5.

[6] R. K. Halder, M. N. Uddin, Md. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J Big Data*, vol. 11, no. 1, p. 113, Aug. 2024, doi: 10.1186/s40537-024-00973-y.

[7] E. Kandić, A. Akagic, and M. Bohlouli, "Exploring Convolutional Autoencoder Efficacy in Noise Removal for Image Processing and Computer Vision: A Study Using the MNIST Dataset," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, Jul. 2024, pp. 1275–1280. doi: 10.1109/CoDIT62066.2024.10708626.

[8] J. Mei, "A comparison between CNN and leNet in fashion Mnist dataset," 2024, p. 040017. doi: 10.1063/5.0223181.

[9] UCI ML repository, "Optical Recognition of Handwritten Digits - UCI Machine Learning Repository." Accessed: May 29, 2025. [Online]. Available: https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits

[10] S. V. Razavi-Termeh, A. Sadeghi-Niaraki, S. Razavi, and S.-M. Choi, "Enhancing flood-prone area mapping: fine-tuning the K-nearest neighbors (KNN) algorithm for spatial modelling," *Int J Digit Earth*, vol. 17, no. 1, Dec. 2024, doi: 10.1080/17538947.2024.2311325.

[11] M. I. Fidatama, F. Bimantoro, G. S. Nugraha, B. Irmawati, and R. Dwiyansaputra, "Recognition of Bima script handwriting patterns using the local binary pattern feature extraction method and K-nearest neighbour classification method," 2023, p. 020009. doi: 10.1063/5.0111770.

[12] A. Susanto, I. U. W. Mulyono, C. A. Sari, E. H. Rachmawanto, and R. R. Ali, "Javanese Character Recognition Based on K-Nearest Neighbor and Linear Binary Pattern Features," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, Sep. 2022, doi: 10.22219/kinetik.v7i3.1491.

[13] F. E. Batool *et al.*, "Offline signature verification system: a novel technique of fusion of GLCM and geometric features using SVM," *Multimed Tools Appl*, vol. 83, no. 5, pp. 14959–14978, Apr. 2020, doi: 10.1007/s11042-020-08851-4.

p-ISSN: 2803-1507 e-ISSN: 2803-1531

[14] I. W. A. S. Darma, "Handwritten Balinesse Character Recognition using K-Nearest Neighbor," May 22, 2018. doi: 10.31227/osf.io/z6m8u.

[15] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah, "Multi-class K-support Vector Nearest Neighbor for Mango Leaf Classification," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 4, p. 1826, Aug. 2018, doi: 10.12928/telkomnika.v16i4.8482.

[16] A. S. Wijaya, N. Chamidah, and M. M. Santoni, "Pengenalan Karakter Tulisan Tangan Dengan K-Support Vector Nearest Neighbor," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 9, no. 1, p. 33, Apr. 2019, doi: 10.22146/ijeis.38729.

[17] E. Prasetyo, R. D. Adityo, and R. Purbaningtyas, "Classification of Segmented Milkfish Eyes using Cosine K-Nearest Neighbor," in *2019 2nd International Conference on Applied Information Technology and Innovation (ICAITI)*, IEEE, Sep. 2019, pp. 93–98. doi: 10.1109/ICAITI48442.2019.8982124.

[18] P. H. Jain, V. Kumar, J. Samuel, S. Singh, A. Mannepalli, and R. Anderson, "Artificially Intelligent Readers: An Adaptive Framework for Original Handwritten Numerical Digits Recognition with OCR Methods," *Information*, vol. 14, no. 6, p. 305, May 2023, doi: 10.3390/info14060305.

[19] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[20] D. Impedovo and G. Pirlo, "Zoning methods for handwritten character recognition: A survey," *Pattern Recognit*, vol. 47, no. 3, pp. 969–981, Mar. 2014, doi: 10.1016/j.patcog.2013.05.021.

[21] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognit*, vol. 37, no. 2, pp. 265–279, Feb. 2004, doi: 10.1016/S0031-3203(03)00224-3.

[22] K. Korovai, D. Zhelezniakov, O. Yakovchuk, O. Radyvonenko, N. Sakhnenko, and I. Deriuga, "Handwriting Enhancement: Recognition-Based and Recognition-Independent Approaches for On-Device Online Handwritten Text Alignment," *IEEE Access*, vol. 12, pp. 99334–99348, 2024, doi: 10.1109/ACCESS.2024.3412433.

[23] C. Reddy, S. Mallesh, R. Rao, R. Sivan, and P. B. Pati, "Comparison of GLCM and DCT features for Malayalam Palm Leaf Character Recognition," in *2024 5th International Conference for Emerging Technology (INCET)*, IEEE, May 2024, pp. 1–5. doi: 10.1109/INCET61516.2024.10592989.