Jurnal Teknik Indonesia

Volume 5 Nomor 3 Juli 2026

https://jti.publicascientificsolution.com/index.php/rp

Analisis Kombinasi *K-Nearest Neighbor* (KNN) dan *K-Means* pada Klasifikasi Penyakit Daun Padi dengan Menggunakan Metode Segmentasi Citra

Rizal Afandi* , Kartini, Retno Mumpuni
Universitas Pembangunan Nasional, Indonesia
Email: 19081010146@student.upnjatim.ac.id* , kartini.if@upnjatim.ac.id, retnomumpuni.if@upnjatim.ac.id

Keywords:

K-Means; K-Nearest Neighbor (KNN); Image Classification; Rice Leaf Disease; Image Segmentation.

Abstract

Rice leaf disease is one of the main factors causing decreased rice productivity and potentially threatening food security in Indonesia. The manual identification process of plant diseases is often time-consuming and prone to errors. This study aims to implement a combination of K-Means and K-Nearest Neighbor (KNN) algorithms for rice leaf disease classification using image segmentation methods. The dataset used consisted of 5,932 rice leaf images categorized into four disease classes: Bacterial Blight, Blast, Brown Spot, and Tungro. The research stages included image preprocessing through grayscale conversion, resizing, and flattening, followed by segmentation using K-Means and classification using KNN. Testing was conducted using several data split ratios and various K values in the KNN algorithm. The results showed that the 90:10 data split ratio produced the highest accuracy of 88% at K=1. In addition, the model was able to recognize most rice leaf diseases effectively, although some misclassifications still occurred in images with similar visual characteristics. Based on these findings, the combination of K-Means and KNN methods is considered effective and has the potential to be further developed as a fast and efficient digital image-based rice leaf disease detection system.

Kata kunci:

K-Means; K-Nearest Neighbor (KNN); Klasifikasi Citra; Penyakit Daun Padi; Segmentasi Citra.

Abstrak

Penyakit daun padi menjadi salah satu faktor utama yang menyebabkan penurunan produktivitas tanaman padi dan berpotensi mengganggu ketahanan pangan di Indonesia. Proses identifikasi penyakit yang masih dilakukan secara manual sering kali membutuhkan waktu yang lama dan rentan terhadap kesalahan. Penelitian ini bertujuan untuk mengimplementasikan kombinasi algoritma *K-Means* dan *K-Nearest Neighbor* (KNN) pada klasifikasi penyakit daun padi menggunakan metode segmentasi citra. Dataset yang digunakan terdiri dari 5.932 citra daun padi dengan empat kategori penyakit, yaitu *Bacterial Blight, Blast, Brown Spot*, dan *Tungro*. Tahapan penelitian meliputi preprocessing citra berupa *grayscale, resizing*, dan *flattening*, kemudian dilanjutkan dengan segmentasi menggunakan K-Means serta klasifikasi menggunakan KNN. Pengujian dilakukan dengan beberapa rasio pembagian data dan variasi nilai K pada algoritma KNN. Hasil penelitian menunjukkan bahwa rasio data 90:10 menghasilkan akurasi terbaik sebesar 88% pada nilai K=1. Selain itu, model mampu mengenali sebagian besar jenis penyakit daun padi dengan baik

meskipun masih terdapat kesalahan klasifikasi pada citra yang memiliki karakteristik visual serupa. Berdasarkan hasil tersebut, kombinasi metode K-Means dan KNN dinilai efektif dan berpotensi dikembangkan lebih lanjut sebagai sistem deteksi penyakit daun padi berbasis citra digital yang cepat dan efisien.

PENDAHULUAN

Tanaman padi menjadi sumber pangan utama bagi sebagian besar penduduk di Asia, khususnya di Indonesia, di mana beras merupakan komoditas pokok yang mendukung kebutuhan nutrisi masyarakat. Sebagai tanaman yang dibudidayakan secara luas di wilayah dataran rendah, padi memiliki nilai strategis dalam ketahanan pangan dan stabilitas ekonomi masyarakat agraris. Tanaman padi juga menunjukkan ketahanan terhadap berbagai kondisi ekstrem seperti perubahan cuaca, serangan hama dan penyakit tanaman, yang merupakan tantangan signifikan dalam budidaya padi. Meskipun varietas padi telah berkembang melalui proses adaptasi dan seleksi selama ribuan tahun, produktivitas tanaman ini masih sering terancam oleh berbagai jenis penyakit. Menurut laporan dari International Rice Research Institute (IRRI), serangan hama dan penyakit mengakibatkan kerugian panen hingga 37% per tahun di kawasan Asia, yang tentunya berpengaruh besar terhadap ketersediaan pangan di wilayah tersebut (Rekayasa et al., 2022a). Untuk mengurangi dampak tersebut, pengembangan teknologi satelit resolusi tinggi dan indeks vegetasi untuk memantau kondisi tanaman padi memberikan peluang besar dalam meningkatkan produktivitas dan manajemen budidaya tanaman (Vikram & Sarkar, 2022).

Namun, tantangan dalam mengidentifikasi penyakit pada tanaman padi di lapangan masih menjadi kendala signifikan, terutama bagi petani yang tidak memiliki akses atau pengetahuan mendalam mengenai gejala penyakit tanaman. Proses identifikasi penyakit padi yang dilakukan secara manual mengandalkan keahlian dan pengamatan visual terhadap gejala penyakit, seperti perubahan warna dan tekstur daun. Hal ini menyebabkan proses identifikasi menjadi kurang efisien dan sering kali rentan terhadap kesalahan, terutama pada tahap awal perkembangan penyakit yang gejalanya tidak terlihat jelas (Azzahra Nasution et al., 2019). Ketidaktepatan dalam identifikasi dapat memperburuk penyebaran penyakit sehingga berpotensi menyebabkan kerugian besar bagi para petani (Zeger et al., 2021). Oleh karena itu, pengembangan teknologi yang mampu mendukung diagnosis penyakit tanaman secara otomatis sangat diperlukan untuk membantu petani dalam mencegah dan menangani penyakit dengan lebih efektif.

Perkembangan teknologi informasi dan komunikasi (TIK) yang semakin pesat menawarkan solusi yang menjanjikan untuk masalah ini, terutama melalui pengenalan objek berbasis citra digital dan analisis data besar (big data). Dalam konteks pertanian, teknologi pengolahan data besar (big data) seperti data mining terbukti efektif dalam proses ekstraksi informasi penting dari jumlah data yang besar. Melalui teknik clustering dan klasifikasi, data mining memungkinkan pengelompokan dan klasifikasi penyakit tanaman berdasarkan kesamaan karakteristik visual, seperti warna, tekstur, dan pola pada daun padi yang terinfeksi (Hakim et al., 2020). Hal ini memungkinkan sistem deteksi untuk mengenali pola-pola penyakit secara otomatis, sehingga mempermudah proses analisis dan klasifikasi. Teknologi ini juga meningkatkan akurasi identifikasi penyakit tanaman dengan cara mengenali pola visual spesifik yang sulit diidentifikasi oleh manusia. Pendekatan ini menawarkan kecepatan dan efisiensi

dalam deteksi penyakit, yang diharapkan dapat memperbaiki sistem manajemen penyakit tanaman yang lebih terstruktur (Worung et al., 2020a).

Di samping itu, penggunaan algoritma pengelompokan dan klasifikasi, seperti K-Means dan K-Nearest Neighbor (K-NN), dapat memperkaya sistem klasifikasi penyakit pada tanaman padi dengan tingkat akurasi yang lebih tinggi. K-Means adalah metode pengelompokan non-hirarki yang berfungsi untuk membagi data ke dalam beberapa kelompok atau cluster berdasarkan kesamaan karakteristik. Dalam konteks pengelolaan penyakit tanaman, K-Means dapat digunakan untuk mengelompokkan berbagai gejala penyakit berdasarkan pola visual pada daun, sehingga dapat mempermudah proses identifikasi penyakit. Algoritma ini bekerja dengan membagi data ke dalam sejumlah cluster, di mana setiap cluster mewakili pola karakteristik tertentu. K-Means sangat berguna dalam proses ini karena kemampuannya dalam menangani data dengan karakteristik visual yang bervariasi, sehingga memudahkan dalam membedakan jenis-jenis penyakit tanaman yang berbeda (Widyanti et al., 2023).

Di samping itu, penggunaan algoritma pengelompokan dan klasifikasi, seperti K-Means dan K-Nearest Neighbor (K-NN), dapat memperkaya sistem klasifikasi penyakit pada tanaman padi dengan tingkat akurasi yang lebih tinggi. K-Means adalah metode pengelompokan non-hirarki yang berfungsi untuk membagi data ke dalam beberapa kelompok atau cluster berdasarkan kesamaan karakteristik (Darmi et al., 2016; Goralski & Tan, 2020). Dalam konteks pengelolaan penyakit tanaman, K-Means dapat digunakan untuk mengelompokkan berbagai gejala penyakit berdasarkan pola visual pada daun, sehingga dapat mempermudah proses identifikasi penyakit (Chen et al., 2023). Algoritma ini bekerja dengan membagi data ke dalam sejumlah cluster, di mana setiap cluster mewakili pola karakteristik tertentu. K-Means sangat berguna dalam proses ini karena kemampuannya dalam menangani data dengan karakteristik visual yang bervariasi, sehingga memudahkan dalam membedakan jenis-jenis penyakit tanaman yang berbeda (Widyanti et al., 2023). Di sisi lain, algoritma K-Nearest Neighbor (K-NN) digunakan sebagai metode klasifikasi yang memanfaatkan data historis atau data latih untuk menentukan kategori suatu objek berdasarkan kedekatan karakteristiknya. Algoritma ini bekerja dengan membandingkan data baru terhadap data yang telah diklasifikasikan sebelumnya sehingga mampu mengidentifikasi jenis penyakit tanaman berdasarkan kemiripan pola visual pada daun padi (Isman et al., 2021; Sandy et al., 2019). Penelitian sebelumnya menunjukkan bahwa K-NN efektif digunakan dalam klasifikasi objek berbasis citra dengan tingkat akurasi yang tinggi, terutama pada data yang memiliki variasi visual dan rentan terhadap noise.

Penelitian terdahulu telah menunjukkan efektivitas metode K-Means dan KNN dalam klasifikasi berbasis citra pada berbagai domain. Suban et al. (2020a) menerapkan K-Means untuk segmentasi citra dalam identifikasi penyakit tanaman dengan hasil yang menjanjikan. Trisnawan et al. (2019) menggunakan KNN untuk klasifikasi objek berbasis citra dengan akurasi yang baik. Premana & Pandhu Wijaya (2022) mengkombinasikan kedua metode tersebut untuk klasifikasi buah mangga, sementara Febrinanto et al. (2018) menerapkan K-Means sebagai metode segmentasi dalam identifikasi penyakit daun jeruk. Meskipun penelitian-penelitian tersebut telah memberikan kontribusi penting, masih terdapat keterbatasan dalam hal jumlah dataset yang digunakan, variasi jenis penyakit yang diklasifikasikan, serta pengujian kombinasi metode segmentasi dan klasifikasi secara komprehensif pada penyakit daun padi. Sebagian besar penelitian sebelumnya menggunakan dataset yang relatif kecil atau

hanya berfokus pada satu atau dua jenis penyakit, sehingga kurang merepresentasikan keragaman penyakit yang sebenarnya terjadi di lapangan.

Kebaruan penelitian ini terletak pada penggunaan dataset yang lebih besar dan beragam dengan total 5.932 citra daun padi yang mencakup empat kategori penyakit, sehingga memberikan representasi yang lebih komprehensif dibandingkan penelitian sebelumnya. Penelitian ini juga menguji secara sistematis pengaruh berbagai rasio pembagian data latih dan data uji serta variasi nilai K pada algoritma KNN untuk menentukan kombinasi parameter yang paling optimal. Selain itu, penelitian ini mengintegrasikan segmentasi citra menggunakan K-Means dengan klasifikasi menggunakan KNN dalam satu kerangka kerja yang terstruktur, serta mengevaluasi performa model menggunakan metrik Silhouette Score dan confusion matrix secara komprehensif.

Berdasarkan latar belakang, penelitian terdahulu, dan gap penelitian yang telah diuraikan, penelitian ini bertujuan untuk mengimplementasikan kombinasi algoritma K-Means dan KNN pada klasifikasi penyakit daun padi menggunakan metode segmentasi citra, menganalisis pengaruh rasio pembagian data terhadap akurasi klasifikasi, serta menentukan nilai K optimal pada algoritma KNN untuk klasifikasi penyakit daun padi. Penelitian ini diharapkan memberikan manfaat, yaitu memperkaya kajian tentang penerapan algoritma machine learning dalam bidang pertanian, khususnya untuk deteksi penyakit tanaman berbasis citra digital, memberikan solusi teknologi yang cepat dan efisien bagi petani dalam mengidentifikasi penyakit daun padi secara akurat, serta menjadi dasar bagi pengembangan sistem deteksi dini penyakit tanaman yang lebih canggih dan terintegrasi. Selain itu, penelitian ini diharapkan mampu mendukung upaya peningkatan produktivitas pertanian dan ketahanan pangan nasional melalui deteksi penyakit yang lebih cepat dan tepat sasaran.

METODE PENELITIAN

Penelitian ini menggunakan jenis penelitian kuantitatif dengan pendekatan eksperimen yang bertujuan untuk mengimplementasikan dan menguji performa kombinasi algoritma K-Means dan K-Nearest Neighbor (KNN) dalam klasifikasi penyakit daun padi berbasis segmentasi citra. Pendekatan eksperimen dipilih karena penelitian ini melibatkan pengujian sistematis terhadap model yang dikembangkan dengan berbagai skenario pengujian, seperti variasi rasio pembagian data latih dan data uji serta variasi nilai K pada algoritma KNN, untuk memperoleh hasil yang terukur dan dapat direplikasi.

Untuk mendapatkan informasi dan data-data yang diperlukan dalam pengumpulan data menggunakan teknik sebagai berikut :

Studi Literatur

Pada tahap ini dilakukan pencarian dan pengumpulan referensi dan literatur. Tahapan studi dilakukan untuk mendapatkan informasi yang digunakan sebagai acuan, tahapan studi literatur melibatkan membaca jurnal ilmiah, buku referensi, penelitian terdahulu, dan sumber lain yang berkaitan dengan topik penelitian. Metode ini menggunakan penelitian literatur dari berbagai bidang ilmu yang berkaitan dengan saran dan alur kerja program :

1. Jenis-jenis penyakit pada tanaman padi
2. Pengolahan citra digital
3. Konsep algoritma K-Nearest Neighbor (K-NN) untuk Klasifikasi penyakit pada daun padi
4. Konsep algoritma K-Nearest Neighbor (K-NN) untuk Klasifikasi penyakit pada daun padi

5. Pemrograman Pyhton

Selain bisa mendapatkan penjelasan terkait konsep-konsep dan definisi dari sebuah metode dan istilah-istilah asing, dari tahapan ini penulis juga bisa menemukan gap dari penelitian sebelumnya dan mendapatkan beberapa opsi cara dalam melakukan perancangan sistem. Berdasarkan uraian diatas, dapat disimpulkan bahwa tahapan studi literatur ini termasuk tahapan yang cukup penting pada proses penelitian.

Pengumpulan Data Set

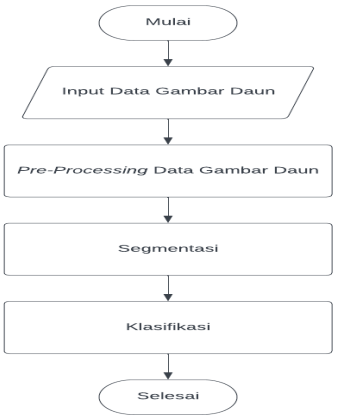
Pada penelitian ini, untuk melakukan pengumpulan data menggunakan data set yang diambil dari website Kaggle dengan nama pemilik user NIRMAL SANKALANA yang berisi tentang macam-macam penyakit yang ada pada daun padi. Data tersebut bersifat publik dan terdiri dari 4 jenis data yaitu penyakit Bacterial Blight terdapat 1584 gambar, penyakit Blast terdapat 1440 gambar, penyakit Brown Spot terdapat 1600 gambar, dan penyakit Tungro terdapat 1308 gambar dengan format JPG dengan total dataset berjumlah 5932 data. Sebanyak 5932 data citra daun dikumpulkan dan dibagi menjadi 2000 data uji dan 3932 data latih, terdiri dari masing-masing 500 jenis penyakit daun dengan kategori Bacterial Blight, Blast, Brown Spot, dan Tungro. Kemudian, 3932 data latih juga dibagi menjadi 983 untuk masing-masing kategori (Bacterial Blight, Blast, Brown Spot, dan Tungro). Setelah itu melakukan cropping sesuai dengan background alat bantu. Berikut merupakan contoh sampel data yang akan digunakan :

Gambar 1. Data Sampel yang digunakan

Sumber: Dataset Kaggle (NIRMAL SANKALANA)

Perancangan Sistem

Perancangan diperlukan dalam penelitian ini untuk melakukan segmentasi citra menggunakan algoritma K-Means dan proses klasifikasi penyakit daun padi. Tahap pertama yaitu melakukan input data gambar penyakit daun padi. Tahap kedua melakukan pre-processing data gambar penyakit daun padi menggunakan metode-metode pengolahan citra digital. Tahap ketiga, melakukan segmentasi penyakit daun padi menggunakan algoritma K-Means. Dan tahap keempat melakukan klasifikasi menggunakan algoritma K-NN¬.



Gambar 2. Flowchart Perancangan Program

Sumber: Hasil Perancangan Penelitian (2025)

Pre-prosessing Citra

Pada tahap preprocessing citra, dataset yang telah dikumpulkan sebelumnya akan diproses untuk mempersiapkan data sebelum dilakukan segmentasi dan klasifikasi. Tahapan ini bertujuan untuk menyederhanakan data citra, mengurangi kompleksitas, serta memastikan bahwa data yang digunakan memiliki format yang seragam sehingga dapat diproses dengan baik oleh algoritma. Dalam penelitian ini, preprocessing citra dilakukan melalui beberapa tahapan, yaitu konversi citra ke grayscale, resizing citra, serta transformasi citra menjadi vektor satu dimensi. Pada tahap grayscale, citra daun padi yang awalnya berformat warna (RGB) dikonversi menjadi citra grayscale menggunakan fungsi cv2.cvtColor dari library OpenCV. Setelah proses grayscale, tahap berikutnya adalah resizing citra untuk menyeragamkan ukuran seluruh data pada dataset menjadi 64×64 piksel menggunakan fungsi cv2.resize. Tahap selanjutnya adalah transformasi data atau flattening, yaitu mengubah data citra menjadi bentuk satu dimensi agar dapat diproses oleh algoritma K-Means dan KNN.

Segmentasi Citra Menggunakan K-Means

Setelah tahap preprocessing selesai dilakukan, langkah selanjutnya adalah segementasi citra menggunakan algoritma K-Means. Pada peneltian ini, K-Means digunakan untuk mengelompokkan data citra berdasarkan kemiripan karakteristik yang dimiliki. Proses segmentasi dilakukan dengan memnetukan jumlah cluster sebanyak 4, yang disesuaikan dengan jumlah kelas penyakit daun padi yang digunakan dalam penelitian ini, yaitu Bacterial Blight, Blast, Brown Spot, dan Tungro. Algoritma K-Means akan mengelompokkan data berdasarkan jarak terdekat terhadap centroid yang telah ditentukan.

Dalam implementasinya, data citra yang telah melalui tahap preprocessing akan digunakan sebagai input untuk proses clustering. Hasil dari proses ini berupa pengelompokkan data yang selanjutnya akan digunakan sebagai input pada tahap klasifikasi menggunakan K-Nearest Neighbor (KNN). Dengan adanya proses segmentasi ini, diharapkan data yang memiliki karakteristik serupa dapat dikelompokkan dengan baik sehingga dapat meningkatkan akurasi pada tahap klasifikasi.

Klasifikasi Menggunakan K-Nearest Neighbor (KNN)

Tahap selanjutnya setelah proses segmentasi adalah klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN). Algoritma ini digunakan untuk menentukan kelas dari data citra berdasarkan kedekatan dengan data latih yang telah diketahui labelnya. Dalam peneltian ini, data hasil transformasi dari K-Means akan dihunakan sebagai input pada algoritma KNN.

Proses klasifikasi dilakukan dengan membandingkan jarak antara data uji dengan data latih menggunkan nilai K tertentu. Nilai K dalam penlitian ini diuji dengan beberapa variasi, yaitu dari K = 1 hingga K = 10, untuk menentukan nilai K optimal berdasarkan hasil akurasi terbaik. Semakin kecil nilai K, maka model akan lebi sensitifd terhadap data latih, sedangkan nilai K yang lebih besar menghasilkan keputusan yang lebih umum. Hasil dari proses klasifikasi ini berupa prediksi kelas penyakit daun padi yang kemudian akan dievaluasi pada tahap pengujian.

Implementasi

Tahap implemntasi merupakan tahap pembangunan sistem berdasarkan perancangan yang telah dilakukan sebelumnya. Pada penelitian ini, implentasi dilakukan menggunakan bahasa pemograman Python dengan memanfaatkan beberapa library seperti OpenCV, NumPy, dan Scikit-learn.

Seluruh tahapan mulai dari preprocessing citra, segmentasi menggunakan K-Means, hingga klasifikasi menggunakan KNN diintegrasikan dalam satu sitem yang mampu melakukan identifkasi penyakit daun padi secara otomatis.

Pengujian dan Analisis

Tahap pengujian yang pertama dilakukan dengan memperhatikan nilai Scale Factor pada proses rescaling. Untuk menentukan nilai Scale Factor terbaik, dilakukan dengan mencari hasil akurasi terbaik dalam beberapa percobaan nilai Scale Factor. Kedua, menggunakan metode Silhouette Coefficient untuk menentukan nilai terbaik dari jumlah cluster pada segmentasi menggunakan K-Means. Ketiga, Melakukan pengujian untuk mendapatkan nilai K optimal dalam pemrosesan dengan menggunakan algoritma K-NN. Untuk mendapatkan nilai K optimal dilakukan pengujian dengan metode mencari hasil akurasi terbaik dalam percobaan beberapa nilai K. Yang terakhir melakukan analisis pada nilai akurasi program dari hasil identifikasi penyakit menggunakan algoritma K-NN yang dihasilkan melalui proses segmentasi citra menggunakan algoritma K-Means.

HASIL DAN PEMBAHASAN

Implementasi Program

Subbab ini menjelaskan implementasi program yang menggunkan K-Means dan K-Nearest Neighbor (KNN) untuk indentifikasi gambar penyakit daun padi. Hal ini akan dijelaskan pada sub-bab sesuai proses yang dilakukan dibawah ini.

Persiapan Dataset

Data yang digunakan pada penilitan ini menggunakan data yang berasal dari Kaggle. Data tersebut merupakan data publik yang terdiri dari empat kelas, yaitu Bacterial blight, blast, brownspot, dan tungro. Dataset tersebut memiliki total gambar sebanyak penyakit Bacterial Blight terdapat 1584 gambar, penyakit Blast terdapat 1440 gambar, penyakit Brown Spot terdapat 1600 gambar, dan penyakit Tungro terdapat 1308 gambar dengan format JPG dengan total dataset berjumlah 5932 data. Data gambar tersebut kemudian dikumpulkan menjadi satu folder yang telah dibedakan perkelasnya lalu diunggah ke Google Drive untuk memudahkan proses-proses yang akan dilkukan selanjutnya seperti yang terlihat pada gambar 4.1

Gambar 3. Folder Dataset

Sumber: Dokumentasi Penelitian (2025)

Praproses Data

Dalam tahapan pra-proses data, dataset yang telah dikumpulkan sebelumnya dilakukan proses pre processing data. Proses ini dilakukan dengan menggunggah data, merubah bentuk data citra dan melakukan ekstrasi fitur ke dalam format Grayscale, setelah itu dataset citra akan dibagi sesuai dengan kelas.

```
# Mount Gdrive dan Memuat Dataset
from google.colab import drive
drive.mount('/content/drive')

folder_path = "/content/drive/MyDrive/Skripsi/Sample"
```

Gambar 4. Kode program memuat data set (Kode program 1)

Kode Program 1, menentukan direktori dataset citra disimpan dalam hal ini diatur ke dalam penyimpanan yang telah disiapkan pada google drive "/content/drive/MyDrive/Skripsi/Sample". Dalam source code diatas terdapat code untuk mengkoneksikan antara google colab dengan google drive sehingga data yang dipanggil akan termuat atau diproses lebih lanjut.

```
# Fungsi untuk memproses gambar (grayscale, resizing, flattening)
def load_and_process_images(folder, num_images, img_size=(64, 64)):

    images = []

    filenames = os.listdir(folder)[:num_images]

    for filename in tqdm(filenames, desc=f"Processing
{folder.split('/')[-1]}"):

        img = cv2.imread(os.path.join(folder, filename))

        if img is not None:

            gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

            resized_img = cv2.resize(gray_img, img_size)

            flattened_img = resized_img.flatten()

            images.append(flattened_img)
```

Gambar 5. Kode program praproses data (Kode program 2)

Kode program 2 menjadi bagian penting dalam langkah-langkah memuat dataset citra yang berformat .jpg. Dalam potongan porgram diatas menggunakan fungsi ‘load_and_process_images’ untuk memuat dan memproses gambar dalam grayscale dari direktori yang ditentukan. Fungsi ini tidak hanya memuat gambar, tetapi juga mengonversi gambar menjadi grayscale, mengubah ukurannya, serta meratakan gambar untuk digunakan

dalam analisis lebih lanjut. Fungsi ini menerima tiga parameter, yaitu folder sebagai jalur direktori tempat gambar-gambar disimpan, num_images untuk menentukan jumlah gambar yang akan dimuat, dan img_size untuk menentukan ukuran gambar setelah diubah ukurannya (default-nya adalah 64x64 piksel).

Proses dimulai dengan inisialisasi list kosong images yang akan menyimpan gambar-gambar yang telah diproses. Fungsi kemudian mengambil sejumlah nama file sesuai dengan num_images dari direktori yang ditentukan oleh folder. Selama iterasi, gambar di setiap file dibaca menggunakan cv2.imread. Jika gambar berhasil dimuat, gambar tersebut diubah menjadi grayscale menggunakan cv2.cvtColor dengan flag cv2.COLOR_BGR2GRAY. Gambar grayscale ini kemudian diubah ukurannya sesuai dengan parameter img_size menggunakan cv2.resize. Selanjutnya, gambar yang telah diubah ukurannya diratakan menjadi vektor satu dimensi menggunakan metode flatten. Gambar yang telah diratakan ini kemudian ditambahkan ke dalam list images.

Untuk memberikan umpan balik kepada pengguna selama proses berlangsung, iterasi dibungkus dengan fungsi tqdm, yang menampilkan progress bar. Setelah semua gambar selesai diproses, list images dikonversi menjadi array NumPy dan dikembalikan sebagai output dari fungsi.

Setelah seluruh gambar terpilih dimuat, list images dikembalikan sebagai output dari fungsi. Konversi gambar ke format grayscale ini dilakukan untuk menyederhanakan data dan mengurangi kebutuhan memori.

```
# Memuat dan Memporses data
class1_images = load_and_process_images(os.path.join(folder_path,
"Bacterial blight"), 1584)
class2_images = load_and_process_images(os.path.join(folder_path,
"Blast"), 1440)
class3_images = load_and_process_images(os.path.join(folder_path,
"Brownspot"), 1600)
class4_images = load_and_process_images(os.path.join(folder_path,
"Tungro"), 1308)
```

Gambar 6. Kode program memuat dataset dan memproses data (Kode program 3)

Kode program 3 menjadi bagian penting dalam langkah-langkah memuat dataset dengan format grayscale. Dilakukan proses pemuatan citra dalam grayscale untuk empat kelas penyakit daun padi, yaitu Bacterial Blight, Blast, Brown Spot, dan Tungro dimuat dan diproses menggunakan fungsi load_and_process_images. Fungsi ini tidak hanya memuat gambar dari direktori yang sesuai untuk masing-masing kelas, tetapi juga melakukan serangkaian operasi pemrosesan gambar, seperti konversi ke skala abu-abu, pengubahan ukuran gambar menjadi 64x64 piksel, dan perataan gambar menjadi vektor satu dimensi.

Gambar-gambar dari kelas Bacterial Blight, Blast, Brown Spot, dan Tungro masing-masing berjumlah 1584, 1440, 1600, dan 1308 gambar. Setiap kelas gambar diproses secara terpisah dan disimpan dalam variabel class1_images, class2_images, class3_images, dan class4_images. Fungsi load_and_process_images memastikan bahwa setiap gambar yang dimuat dari direktori terkait diubah menjadi grayscale, diubah ukurannya, dan diratakan menjadi vektor satu dimensi sebelum disimpan.

```
# Menggabungkan data dan label

all_images = np.vstack((class1_images, class2_images, class3_images,
class4_images))

all_labels = np.array([0]*1584 + [1]*1440 + [2]*1600 + [3]*1308)
```

Gambar 7. Menggabungkan data dan label (Kode program 4)

Kode program 4 menjadi bagian penting dalam proses pemuatan dan pemrosesan gambar dari empat kelas penyakit daun padi, yaitu Bacterial Blight, Blast, Brown Spot, dan Tungro. Proses ini dilakukan melalui fungsi load_and_process_images, yang memuat gambar dari setiap kelas, mengonversinya menjadi grayscale, mengubah ukurannya menjadi 64x64 piksel, dan meratakannya menjadi vektor satu dimensi. Dengan langkah-langkah ini, setiap gambar dari masing-masing kelas diolah sedemikian rupa sehingga siap untuk dianalisis secara lebih lanjut.

Setelah gambar dari setiap kelas berhasil diproses, fungsi 'all_labels' berperan dalam menggabungkan seluruh gambar menjadi satu kesatuan dan memberikan label pada setiap kelas gambar. Gambar-gambar dari keempat kelas tersebut digabungkan menjadi satu array besar menggunakan np.vstack. Label numerik kemudian diberikan pada masing-masing gambar sesuai dengan kelasnya menggunakan np.array, dengan label 0 untuk Bacterial Blight, 1 untuk Blast, 2 untuk Brown Spot, dan 3 untuk Tungro.

Proses Data

Dalam tahapan proses data, dataset yang sebelumnya telah melalui pra-proses data akan proses selanjutnya. Proses ini diawali dengan pengujian data, menerapkan model k-means clustering, menerapkan model k-nn clustering, serta memvisualisasikan hasil prediksi.

```
# Fungsi untuk menjalankan berbagai skenario pengujian

def run_all_scenarios(all_images, all_labels):

    results = []

    split_ratios = [0.1, 0.2, 0.3, 0.4]



    for test_size in split_ratios:

        train_images, test_images, train_labels, test_labels =
train_test_split(

            all_images, all_labels, test_size=test_size,
random_state=42)



        silhouette_score_train = []
```

Gambar 8. Kode Program 5 Menjalankan berbagai skenario yang akan diuji

Kode program 5 menjadi bagian penting dalam menjalankan berbagai skenario pengujian pada data gambar yang telah diproses. Fungsi run_all_scenarios menguji model dengan menggunakan berbagai rasio pembagian data latih dan data uji yang ditentukan dalam variabel split_ratios (0.1, 0.2, 0.3, dan 0.4). Untuk setiap rasio pembagian, data dibagi menjadi

data latih dan data uji menggunakan train_test_split. Variabel train_images dan test_images menyimpan gambar-gambar yang digunakan untuk training dan pengujian, sementara train_labels dan test_labels menyimpan label yang sesuai.

Selanjutnya, fungsi ini menginisialisasi dua list, silhouette_score_train dan silhouette_score_test, untuk menyimpan nilai Silhouette Score dari data latih dan data uji. Untuk setiap nilai K dari 1 hingga 10, model K-Means digunakan untuk mengelompokkan data latih menjadi 4 klaster. Silhouette Score dihitung untuk data latih dan data uji berdasarkan hasil klasterisasi. Model K-Nearest Neighbors (KNN) kemudian dilatih menggunakan hasil transformasi K-Means dari data latih dan diuji pada data uji. Akurasi prediksi dihitung dan semua hasil pengujian disimpan dalam list results. Selain itu, fungsi visualize_predictions dipanggil untuk menampilkan gambar hasil prediksi, memberikan visualisasi yang jelas dari performa model.

```
# K-Means Clustering

kmeans = KMeans(n_clusters=4, random_state=42)

kmeans.fit(train_images)

# Silhouette Score

silhouette_score_train.append(silhouette_score(train_images,
kmeans.labels_))

silhouette_score_test.append(silhouette_score(test_images,
kmeans.predict(test images)))
```

Gambar 9. Kode Program 6 Penerapan K-Means clustering

Kode program 6 menjelaskan penerapan metode K-Means Clustering pada data latih. Di bagian ini, model K-Means diinisialisasi dengan parameter n_clusters=4 untuk mencocokkan jumlah klaster dengan jumlah kelas penyakit daun padi. Model ini kemudian dilatih menggunakan data latih (train_images). Setelah klasterisasi selesai, nilai Silhouette Score dihitung untuk mengukur kualitas klaster pada data latih dan data uji. Nilai Silhouette Score ini memberikan informasi tentang seberapa baik data latih dan uji di-klasterisasi, dengan hasil disimpan dalam list silhouette_score_train dan silhouette_score_test. Nilai ini berguna untuk menilai kualitas klasterisasi yang dilakukan oleh model K-Means.

```
# K-Nearest Neighbors (KNN)

knn = KNeighborsClassifier(n_neighbors=k)

knn.fit(kmeans.transform(train_images), train_labels)

predictions = knn.predict(kmeans.transform(test_images))


accuracy = accuracy_score(test_labels, predictions)

results.append({

     "split_ratio": f"{(1-test_size)*100:.0f}:{test_size*100:.0f}",

     "K": k,

     "accuracy": accuracy,

     "silhouette_score_train": silhouette_score_train[-1],

     "silhouette_score_test": silhouette_score_test[-1]
```

Gambar 10. Kode Program 7 Penerapan K-Nearest Neighbor (KNN)

Kode program 7 berfungsi untuk menerapkan metode K-Nearest Neighbors (KNN) pada data setelah proses klasterisasi K-Means. Setelah model K-Means dilatih, data latih yang telah ditransformasikan oleh K-Means digunakan untuk melatih model K-Nearest Neighbor (KNN) dengan berbagai nilai K (dari 1 hingga 10). Model K-Nearest Neighbor (KNN) dilatih dengan data latih yang telah diklasterisasi dan diuji pada data uji yang juga telah ditransformasikan dengan K-Means. Akurasi dari prediksi model K-Nearest Neighbor (KNN) dihitung menggunakan accuracy_score, dan hasilnya disimpan dalam list results. Penilaian akurasi ini memberikan gambaran tentang seberapa baik model K-Nearest Neighbor (KNN) dalam mengklasifikasikan data uji setelah data tersebut dikelompokkan oleh K-Means.

```
# Visualization of predictions

visualize_predictions(test_images, test_labels, predictions,
test_size, k)

    return pd.DataFrame(results)
```

Gambar 11. Kode Program 8 Visualisasi Prediksi

Kode program 8 melibatkan visualisasi hasil prediksi untuk memberikan gambaran visual mengenai kinerja model. Fungsi visualize_predictions digunakan untuk menampilkan gambar hasil prediksi bersama dengan label asli pada setiap skenario pengujian. Visualisasi ini dilakukan untuk setiap kombinasi rasio pembagian data dan nilai K, memudahkan evaluasi dan interpretasi performa model K-Means dan K-Nearest Neighbor (KNN). Gambar-gambar yang ditampilkan memberikan insight langsung tentang kualitas prediksi dan kesesuaian label yang diprediksi dengan label asli. Hasil visualisasi ini, bersama dengan data hasil pengujian yang telah dikumpulkan, disimpan dalam bentuk DataFrame pandas untuk analisis lebih lanjut.

Hasil Pengujian

Pengujian dilakukan untuk mengevaluasi performa model kombinasi algoritma K-Means dan K-Nearest Neighbor (KNN) dalam klasifikasi penyakit daun padi. Proses evaluasi

melibatkan beberapa rasio pembagian data latih dan data uji, yaitu 90:10, 80:20, 70:30, dan 60:40, serta pengujian dengan berbagai nilai parameter K pada algoritma K-Nearest Neighbor (KNN), mulai dari K = 1 hingga K = 10, untuk menentukan nilai K yang optimal. Pengujian ini bertujuan untuk menilai kemampuan algoritma dalam mengklasifikasikan data secara akurat dan menganalisis kesalahan klasifikasi yang terjadi melalui confusion matrix.

Hasil pengujian terdiri dari dua komponen utama, yaitu confusion matrix dan visualisasi prediksi. Confusion matrix digunakan untuk memberikan gambaran kuantitatif mengenai performa model, sedangkan visualisasi prediksi menyajikan contoh data secara visual, lengkap dengan label asli dan hasil prediksi model. Evaluasi ini diharapkan dapat memberikan wawasan mengenai kombinasi parameter dan rasio data yang menghasilkan performa terbaik untuk klasifikasi penyakit daun padi.

1. Ringkasan Akurasi Seluruh Skenario

Tabel berikut menyajikan akurasi model untuk setiap skenario dengan variasi nilai K pada masing-masing rasio pembagian data:

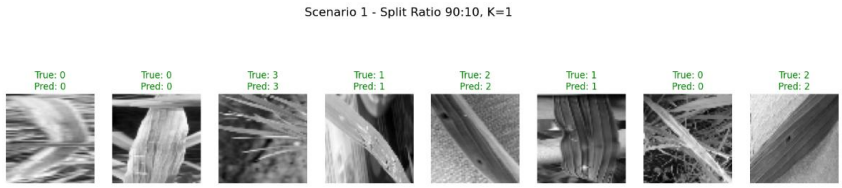
Tabel 1. Ringkasan Akurasi Model pada Seluruh Skenario Pengujian

Rasio Data	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10	Akurasi Terbaik
90:10	88%	86%	86%	82%	84%	81%	85%	84%	84%	82%	88% (K=1)
80:20	48%	44%	44%	36%	38%	34%	39%	35%	36%	32%	48% (K=1)
70:30	27%	22%	22%	19%	20%	18%	20%	18%	19%	18%	27% (K=1)
60:40	45%	42%	42%	38%	38%	35%	36%	33%	34%	31%	45% (K=1)

Sumber: Hasil Pengujian Penelitian (2025)

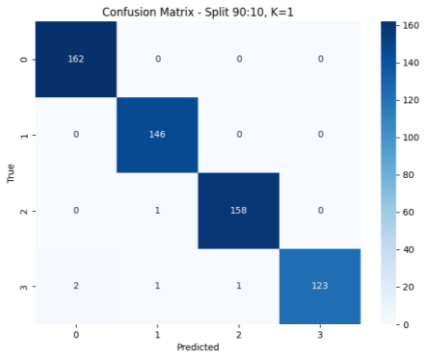
2. Pengujian dengan Rasio Data 90:10

Pada kelompok skenario rasio 90:10 (skenario 1–10), model dilatih dengan 90% data dan diuji dengan 10% data. Pengujian dilakukan untuk nilai K=1 hingga K=10. Gambar 4.2 dan 4.3 menampilkan hasil prediksi dan confusion matrix pada skenario terbaik (K=1) dengan akurasi 88%.



Gambar 12. Hasil Prediksi Skenario 1 (Rasio 90:10, K=1)

Sumber: Hasil Pengujian Penelitian (2025)



Gamba 13. Confusion Matrix Skenario 1 (Rasio 90:10, K=1)

Sumber: Hasil Pengujian Penelitian (2025)

Pada skenario terbaik (K=1), model berhasil memprediksi dengan benar: Kelas 0 (Bacterial Blight) sebanyak 162 data, Kelas 1 (Blast) sebanyak 146 data, Kelas 2 (Brown Spot) sebanyak 158 data, dan Kelas 3 (Tungro) sebanyak 123 data. Kesalahan prediksi minimal terutama terjadi pada Kelas 3, dengan hanya 4 data salah klasifikasi.

Ringkasan hasil confusion matrix untuk seluruh skenario rasio 90:10 disajikan dalam tabel berikut:

Tabel 2. Ringkasan Confusion Matrix – Rasio 90:10

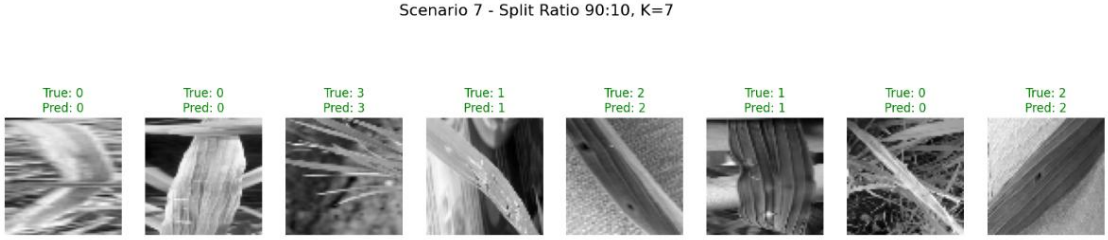
Skenario	K	Kls 0 Benar	Kls 1 Benar	Kls 2 Benar	Kls 3 Benar	Total Salah	Akurasi
1	1	162	146	158	123	4	88%
2	2	162	143	156	122	8	86%
3	3	160	141	156	122	11	86%
4	4	160	135	145	121	29	82%
5	5	159	134	144	121	31	84%
6	6	159	126	140	117	42	81%
7	7	152	121	140	122	37	85%
8	8	152	119	140	120	37	84%
9	9	150	119	133	121	44	84%
10	10	151	115	131	121	42	82%

Sumber: Hasil Pengujian Penelitian (2025)

Analisis: Pada rasio 90:10, model mencapai akurasi tertinggi 88% pada K=1. Akurasi cenderung menurun seiring bertambahnya nilai K, dengan penurunan paling signifikan pada K=4 (82%) dan K=6 (81%). Kelas 1 (Blast) konsisten memiliki kesalahan klasifikasi terbanyak karena karakteristik visualnya mirip dengan Kelas 0 (Bacterial Blight).

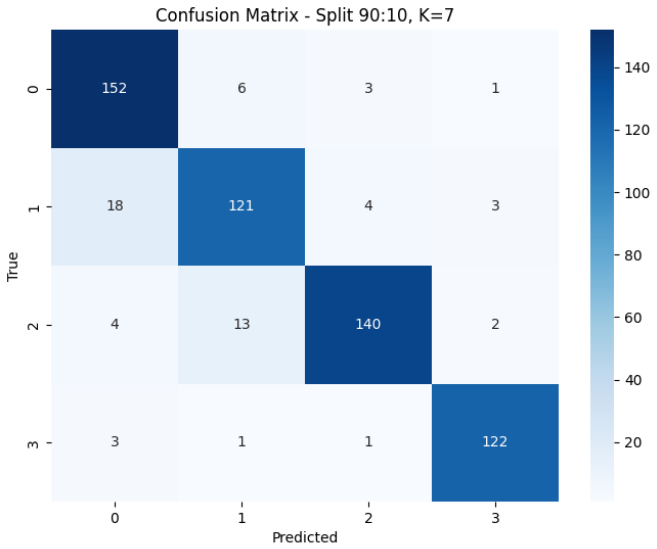
3. Pengujian dengan Rasio Data 80:20

Pada kelompok skenario rasio 80:20 (skenario 11–20), model dilatih dengan 80% data dan diuji dengan 20% data. Gambar 4.4 dan 4.5 menampilkan hasil prediksi dan confusion matrix pada skenario terbaik (skenario 11, K=1) dengan akurasi 48%.



Gambar 14. Hasil Prediksi Skenario 11 (Rasio 80:20, K=1)

Sumber: Hasil Pengujian Penelitian (2025)



Gambar 15. Confusion Matrix Skenario 11 (Rasio 80:20, K=1)

Sumber: Hasil Pengujian Penelitian (2025)

Pada skenario terbaik (K=1), model berhasil mengklasifikasikan dengan benar: Kelas 0 sebanyak 338 data, Kelas 1 sebanyak 268 data, Kelas 2 sebanyak 315 data, dan Kelas 3 sebanyak 254 data.

Tabel 3. Ringkasan Confusion Matrix – Rasio 80:20

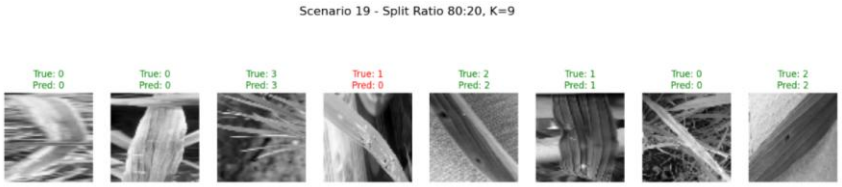
Skenario	K	Kls 0 Benar	Kls 1 Benar	Kls 2 Benar	Kls 3 Benar	Total Salah	Akurasi
11	1	338	268	315	254	10	48%
12	2	338	265	311	246	24	44%
13	3	333	259	311	246	32	44%
14	4	335	241	296	236	73	36%
15	5	333	245	289	241	77	38%
16	6	336	233	280	239	97	34%
17	7	319	227	272	246	101	39%
18	8	321	225	266	241	112	35%
19	9	312	223	267	242	115	36%
20	10	312	216	259	233	131	32%

Sumber: Hasil Pengujian Penelitian (2025)

Analisis: Pada rasio 80:20, akurasi tertinggi hanya mencapai 48% (K=1), jauh lebih rendah dibandingkan rasio 90:10. Tren penurunan akurasi seiring peningkatan K tetap konsisten. Kesalahan klasifikasi paling banyak terjadi pada Kelas 1 (Blast) yang sering salah diklasifikasi sebagai Kelas 0 (Bacterial Blight).

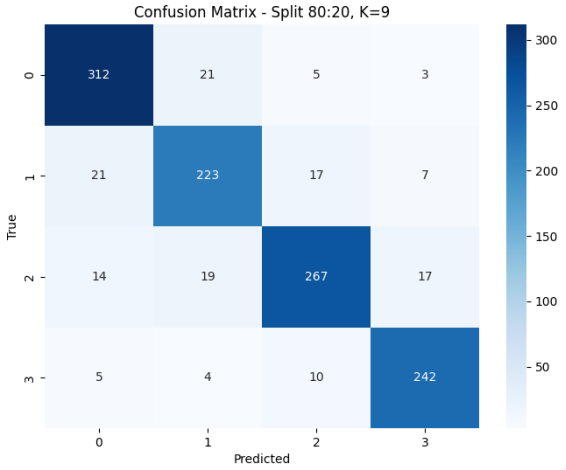
4. Pengujian dengan Rasio Data 70:30

Pada kelompok skenario rasio 70:30 (skenario 21–30), model dilatih dengan 70% data. Gambar 4.6 dan 4.7 menampilkan hasil prediksi dan confusion matrix pada skenario terbaik (skenario 21, K=1) dengan akurasi 27%.



Gambar 16. Hasil Prediksi Skenario 21 (Rasio 70:30, K=1)

Sumber: Hasil Pengujian Penelitian (2025)



Gambar 17. Confusion Matrix Skenario 21 (Rasio 70:30, K=1)

Sumber: Hasil Pengujian Penelitian (2025)

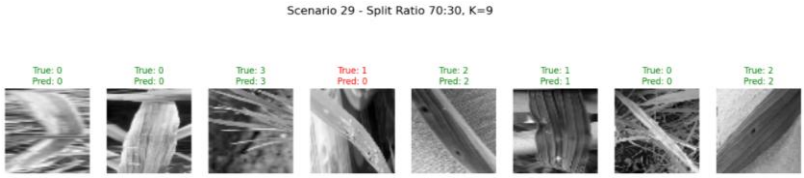
Tabel 4. Ringkasan Confusion Matrix – Rasio 70:30

Skenario	K	Kls 0 Benar	Kls 1 Benar	Kls 2 Benar	Kls 3 Benar	Total Salah	Akurasi
21	1	506	407	459	382	15	27%
22	2	507	384	443	374	57	22%
23	3	493	376	430	376	66	22%
24	4	498	360	414	355	102	19%
25	5	480	355	399	363	119	20%
26	6	485	347	399	357	125	18%
27	7	472	352	390	365	131	20%
28	8	474	326	368	361	157	18%
29	9	446	320	367	363	174	19%
30	10	447	320	363	360	175	18%

Sumber: Hasil Pengujian Penelitian (2025)

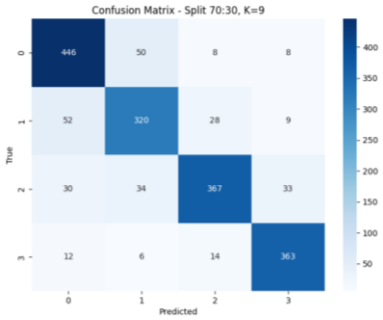
5. Pengujian dengan Rasio Data 60:40

Pada kelompok skenario rasio 60:40 (skenario 31–40), model dilatih dengan hanya 60% data. Gambar 4.8 dan 4.9 menampilkan hasil prediksi dan confusion matrix pada skenario terbaik (skenario 31, K=1) dengan akurasi 45%.



Gambar 18. Hasil Prediksi Skenario 31 (Rasio 60:40, K=1)

Sumber: Hasil Pengujian Penelitian (2025)



Gambar 19. Confusion Matrix Skenario 31 (Rasio 60:40, K=1)

Sumber: Hasil Pengujian Penelitian (2025)

Tabel 5. Ringkasan Confusion Matrix – Rasio 60:40

Skenario	K	Kls 0 Benar	Kls 1 Benar	Kls 2 Benar	Kls 3 Benar	Total Salah	Akurasi
31	1	654	550	615	495	37	45%
32	2	655	521	594	478	64	42%
33	3	630	504	579	484	120	42%
34	4	634	486	547	460	179	38%
35	5	608	486	535	476	182	38%
36	6	612	467	506	469	212	35%
37	7	584	461	502	471	234	36%
38	8	588	443	490	459	266	33%
39	9	577	424	485	464	283	34%
40	10	577	401	480	456	313	31%

Sumber: Hasil Pengujian Penelitian (2025)

Analisis Hasil Pengujian

Tabel 6. Hasil Silhouette Score Seluruh Skenario

NO.	Splitting Data	Silhoutte Score	
		Training	Test
1	60:40	0,0738	0,1018
2	70:30	0,0644	0,0523
3	80:20	0,0563	0,0407
4	90:10	0,0507	0,0408

Sumber: Hasil Pengujian Penelitian (2025)

Silhouette Score digunakan sebagai metrik evaluasi dalam klasterisasi K-Means untuk menilai sejauh mana suatu data berada dalam klaster yang tepat dibandingkan dengan klaster lainnya. Nilai metrik ini berkisar antara -1 hingga 1, di mana nilai yang lebih tinggi menunjukkan klasterisasi yang lebih baik. Pada penelitian ini, dilakukan pengujian dengan empat skenario pembagian data latih dan data uji, yaitu 60:40, 70:30, 80:20, dan 90:10. Hasil evaluasi menunjukkan bahwa skenario 60:40 memiliki nilai Silhouette Score tertinggi pada data training sebesar 0,0738, sedangkan skenario 90:10 memiliki nilai terendah sebesar 0,0507, yang mengindikasikan bahwa semakin besar porsi data latih, kualitas pemisahan klaster tidak selalu meningkat karena kurangnya variasi dalam data. Pada data uji, skenario 60:40 kembali menunjukkan hasil terbaik dengan nilai Silhouette Score sebesar 0,1018, sementara skenario 90:10 memiliki nilai terendah sebesar 0,0408, yang menunjukkan bahwa jumlah data uji yang terlalu sedikit dapat mengurangi efektivitas model dalam membentuk klaster yang jelas.

Meskipun skenario 60:40 memiliki nilai Silhouette Score tertinggi, hasil pengujian menunjukkan bahwa tingkat akurasi model justru lebih tinggi pada skenario 90:10 dibandingkan dengan skenario lainnya. Hal ini menunjukkan bahwa meskipun klasterisasi yang dihasilkan pada skenario 90:10 kurang optimal berdasarkan Silhouette Score, jumlah data latih yang lebih besar mampu meningkatkan performa model dalam proses klasifikasi penyakit daun padi menggunakan K-Nearest Neighbor (KNN). Oleh karena itu, dalam pemilihan rasio pembagian data, perlu dilakukan pertimbangan antara kualitas klasterisasi dan tingkat akurasi model. Jika tujuan utama adalah menghasilkan klaster yang lebih terpisah dan terstruktur dengan baik, maka skenario 60:40 menjadi pilihan yang lebih sesuai. Namun, jika fokus utama adalah meningkatkan akurasi klasifikasi, maka skenario 90:10 lebih direkomendasikan karena jumlah data latih yang lebih besar berkontribusi terhadap peningkatan performa model dalam mengklasifikasikan data baru.

Tabel 7. Hasil Akurasi Seluruh Skenario Pengujian

METODE	DATA SPLIT		skenario	K	akurasi
	LATIH	UJI			
KMEANS & *K-Nearest Neighbor (KNN)*	90%	10%	1	1	88%
			2	2	86%
			3	3	86%
			4	4	82%
			5	5	84%
			6	6	81%
			7	7	85%
			8	8	84%
			9	9	84%
			10	10	82%
	80%	20%	11	1	48%
			12	2	44%
			13	3	44%
			14	4	36%
			15	5	38%
			16	6	34%
			17	7	39%
			18	8	35%
			19	9	36%
			20	10	32%
	70%	30%	21	1	27%
			22	2	22%
			23	3	22%
			24	4	19%
			25	5	20%
			26	6	18%
			27	7	20%
			28	8	18%
			29	9	19%

			30	10	18%
			31	1	45%
			32	2	42%
			33	3	42%
			34	4	38%
			35	5	38%
	60%	40%	36	6	35%
			37	7	36%
			38	8	33%
			39	9	34%
			40	10	31%

Sumber: Hasil Pengujian Penelitian (2025)

Berdasarkan tabel diatas, dilakukan pengujian terhadap kombinasi metode K-Means dan K-Nearest Neighbor (KNN) dengan berbagai skenario pembagian data latih dan data uji, yaitu 90:10, 80:20, 70:30, dan 60:40. Selain itu, dilakukan variasi terhadap nilai K untuk mengevaluasi pengaruhnya terhadap performa klasifikasi. Berdasarkan hasil pengujian, skenario 90:10 menghasilkan akurasi tertinggi, dengan rentang nilai akurasi 83% hingga 88% untuk berbagai nilai K. Hasil ini menunjukkan bahwa dengan jumlah data latih yang lebih besar, model memiliki lebih banyak informasi dalam membangun pola klasifikasi yang lebih akurat, sehingga meningkatkan performa prediksi pada data uji.

Sebaliknya, ketika proporsi data latih semakin berkurang dan data uji meningkat, terjadi penurunan akurasi yang cukup signifikan. Hal ini dapat dilihat pada skenario 60:40, di mana akurasi berkisar antara 30% hingga 44%. Penurunan ini mengindikasikan bahwa dengan jumlah data latih yang lebih sedikit, model menjadi kurang mampu mengenali pola dalam data, sehingga mengurangi kemampuan klasifikasi terhadap data uji. Selain itu, hasil pengujian juga menunjukkan bahwa nilai K yang lebih kecil cenderung menghasilkan akurasi yang lebih tinggi dibandingkan nilai K yang lebih besar. Hal ini dikarenakan nilai K yang kecil lebih responsif terhadap variasi dalam data, sedangkan nilai K yang lebih besar menyebabkan model menjadi terlalu generalisasi, yang pada akhirnya dapat mengurangi ketepatan klasifikasi.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa pemilihan rasio data yang tepat sangat mempengaruhi akurasi model, di mana rasio 90:10 menjadi yang paling optimal, karena memberikan hasil klasifikasi dengan akurasi tertinggi. Namun, rasio yang lebih kecil seperti 60:40 menyebabkan penurunan performa yang cukup drastis. Selain itu, pemilihan nilai K juga memiliki pengaruh penting, di mana nilai K yang terlalu besar dapat menurunkan akurasi karena menyebabkan model terlalu banyak mempertimbangkan tetangga yang mungkin tidak relevan. Oleh karena itu, dalam implementasi metode K-Means dan K-Nearest Neighbor (KNN), perlu dilakukan optimasi terhadap rasio pembagian data serta pemilihan nilai K yang tepat agar diperoleh hasil klasifikasi yang optimal.

KESIMPULAN

Berdasarkan hasil penelitian dan pengujian, kombinasi metode K-Means dan K-Nearest Neighbor (KNN) mampu mengklasifikasikan penyakit daun padi dengan tingkat akurasi yang cukup tinggi. K-Means digunakan untuk segmentasi dan pengelompokan citra berdasarkan

kemiripan fitur visual, sedangkan KNN berperan dalam menentukan kelas penyakit. Hasil pengujian menunjukkan bahwa rasio data 90:10 memberikan akurasi terbaik karena jumlah data training yang lebih banyak meningkatkan stabilitas model, sementara rasio 60:40 menghasilkan akurasi lebih rendah. Nilai K=3 hingga K=5 juga memberikan performa klasifikasi terbaik pada sebagian besar pengujian. Model mampu mengenali penyakit seperti Bacterial Blight, Blast, Brown Spot, dan Tungro dengan baik, meskipun masih terdapat kesalahan klasifikasi pada citra dengan karakteristik visual yang mirip. Secara keseluruhan, kombinasi K-Means dan KNN dinilai efektif serta berpotensi dikembangkan lebih lanjut sebagai sistem otomatis untuk deteksi penyakit daun padi secara cepat dan efisien.

REFERENSI

Azzahra Nasution, D., Khotimah, H. H., & Chamidah, N. (2019). *Perbandingan Normalisasi Data Untuk Klasifikasi Wine Menggunakan Algoritma K-NN* (Vol. 4, Issue 1).

Chen, Y.-A., Hsiao, C.-C., Yang, N., Chiao, M., Peng, W.-H., & Huang, C.-C. (2023). *INDIRECT: Invertible and Discrete Noisy Image Rescaling with Enhancement from Case-dependent Textures*. https://doi.org/10.21203/rs.3.rs-2800974/v1

Darmi, Y., Setiawan, A., Bali, J., Kampung Bali, K., Teluk Segara, K., & Bengkulu, K. (2016). Penerapan Metode Clustering K-Means Dalam Pengelompokan Penjualan Produk. In *Jurnal Media Infotama* (Vol. 12, Issue 2).

Febrinanto, F. G., Dewi, C., Wiratno, A. T., Penelitian, B., Jeruk, T., Subtropika, B., & Litbang Pertanian, B. (2018). *Implementasi Algoritme K-Means Sebagai Metode Segmentasi Citra Dalam Identifikasi Penyakit Daun Jeruk* (Vol. 2, Issue 11). http://j-ptiik.ub.ac.id

Goralski, M. A., & Tan, T. K. (2020). Artificial intelligence and sustainable development. *International Journal of Management Education*, *18*(1). https://doi.org/10.1016/j.ijme.2019.100330

Hakim, L., Kristanto, S. P., Shodiq, M. N., Yusuf, D., Setiawan, W. A., Informatika, T., Banyuwangi, N., Raya, J., & Km, J. (2020). Segmentasi Citra Penyakit Pada Batang Buah Naga Menggunakan Metode Ruang Warna L*A*B*. *Seminar Nasional Terapan Riset Inovatif (Sentrinov) Ke-6 ISAS Publishing Series: Engineering and Science*, *6*(1).

Isman, Andani Ahmad, & Abdul Latief. (2021). Perbandingan Metode KNN Dan LBPH Pada Klasifikasi Daun Herbal. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(3), 557–564. https://doi.org/10.29207/resti.v5i3.3006

Muštra, Mario., Vuković, Josip., & Zovko-Cihlar, Branka. (2020). *Proceedings of ELMAR-2020 : 62nd International Symposium ELMAR-2020 : 14-15 September 2020, Zadar, Croatia*. Croatian Society Electronics in Marine - ELMAR.

Praptana, R. H., Sumardiyono, Y. B., Hartono, S., Trisyono, Y. A., Nyoman Widiarta, D. I., Penelitian, L., Tungro, P., Bulo, J., 101 Lanrang, N., Sidrap, R., & Selatan, S. (2014). Keragaman Virulensi dan Konstruksi Molekuler Virus Tungro pada Padi dari Daerah Endemis. *Penelitian Pertanian Tanaman Pangan*, *33*.

Premana, A., & Pandhu Wijaya, A. (2022). *Klasifikasi Jenis Buah Mangga Menggunakan Metode K-Means Clustering*. *5*.

Rekayasa, K. K., Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Journal of Dinda Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Data Institut Teknologi Telkom Purwokerto*, *2*(1), 37–45. https://www.kaggle.com/tedisetiady/leaf-rice-disease-

Sandy, B., Siahaan, J. K., Permana, P., & Muhathir, *. (2019). Klasifikasi Citra Wayang Dengan Menggunakan Metode k-NN & GLCM. In *Prosiding Seminar Nasional Teknologi Informatika* (Vol. 2).

Suban, I. B., Paramartha, A., Fortwonatus, M., & Santoso, A. J. (2020a). Identification the Maturity Level of Carica Papaya Using the K-Nearest Neighbor. *Journal of Physics: Conference Series*, *1577*(1). https://doi.org/10.1088/1742-6596/1577/1/012028

Suban, I. B., Paramartha, A., Fortwonatus, M., & Santoso, A. J. (2020b). Identification the Maturity Level of Carica Papaya Using the K-Nearest Neighbor. *Journal of Physics: Conference Series*, *1577*(1). https://doi.org/10.1088/1742-6596/1577/1/012028

Trisnawan, A., Harianto, W., Informatika, T., Sains, F., Universitas, D. T., & Malang, K. (2019). Klasifikasi Beras Menggunakan Metode K-Means Clustering Berbasis Pengolahan Citra Digital. In *Jurnal Terapan Sains & Teknologi (RAINSTEK)* | (Vol. 1, Issue 1).

Vikram, N., & Sarkar, N. C. (2022). Rice Bean (Vigna umbellata) – A Potential Legume under Adverse Conditions. *International Journal of Bio-Resource and Stress Management*, *13*(9), 928–934. https://doi.org/10.23910/1.2022.3117a

Widyanti, T., Hilabi, S. S., Hananto, A., Tukino, & Novalia, E. (2023). Implementasi K-Means dan K-Nearest Neighbors pada Kategori Siswa Berprestasi. *Jurnal Informasi Dan Teknologi*, *5*(1), 75–82.

Wijaya, E. S., & Prayudi, Y. (2015). Integrasi Metode Steganografi DCS Pada Image Dengan Kriptografi Blowfish Sebagai Model Anti Forensik Untuk Keamanan Ganda Konten Digital. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*.

Worung, D. T., Sompie, S. R. U. A., & Jacobus, A. (2020). Implementasi K-Means dan K-NN pada Pengklasifikasian Citra Bunga. *Jurnal Teknik Informatika*, *15*(3), 217–222.

Zeger, I., Grgic, S., Vukovic, J., & Sisul, G. (2021). Grayscale Image Colorization Methods: Overview and Evaluation. *IEEE Access*, *9*, 113326–113346. https://doi.org/10.1109/ACCESS.2021.3104515