Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

PENGENALAN POLA PENYAKIT TANAMAN JAGUNG MENGGUNAKAN METODE PRINCIPAL COMPONENT ANALYSIS DAN K-NEAREST NEIGHBOR

Citra Duvita Rahman[1*], Yusuf Ramadhan Nasution[2]
[1,2]Ilmu Komputer, Universitas Islam Negeri Sumatera Utara
email: citraduvitarahman@gmail.com[1*]

Abstrak: Komponen kehidupan yang paling penting di bumi ini adalah tumbuhan salah satu ciptaan Allah, tumbuh-tumbuhan dan buah-buahan banya di dijelaskan secara luar di dalam AL-Qur'an. Tumbuhan bermanfaat untuk bergagai hal,termasuk menyediakan oksigen untuk pernapasan, makanan, bahan bakar, obat-obatan. Penelitian ini bertujuan untuk mengenali pola penyakit pada tanaman jagung menggunakan metode Principal Component Analysis (PCA) dan K-Nearest Neighbor (KNN). Penyakit pada tanaman jagung dapat menurunkan produktivitas dan kualitas hasil panen, sehingga diperlukan metode yang efektif untuk mengidentifikasi penyakit secara dini. PCA digunakan untuk mereduksi dimensi data yang tinggi tanpa kehilangan informasi yang signifikan, sehingga mempermudah proses pengenalan pola. Setelah itu, algoritma KNN diterapkan untuk mengklasifikasikan jenis penyakit berdasarkan pola yang telah terbentuk. Dengan menghitung hasil klasifikasi menggunakan algoritma KNN dengan total data 94, sebagai perhitungan manual digunakan 94 data yaitu, data latih 80% berjumlah 72, data uji 20% berjumlah 22 diperoleh acuracy 77.27%, precision 100%, recall 100%, dan f1-score 100%. Dengan menggunakan algoritma KNN membantu klasifikasi, data berupa attribut dan label dilakukan hold out data latih dan data uji, dimana pelabelan data latih harus ditentukan diawal berupa keterangan kategorik yaitu penyakit Puccinia Polysora, Bipolar Maydis Syn dan Helminthosporium Turcicum.

Kata Kunci : PCA, *K-Nearest Neighbor*, Penyakit tanaman jagung, Pengurangan dimensi, Klasifikasi penyakit

Abstract: *The most important component of life on this earth is plants, one of Allah's creations, many plants and fruits are described externally in the Al-Qur'an. Plants are useful for various things, including providing oxygen for breathing, food, fuel, medicine. This research aims to identify disease patterns in corn plants using the Principal Component Analysis (PCA) and K-Nearest Neighbor (KNN) methods. Diseases in corn plants can reduce productivity and quality of crops, so effective methods are needed to identify diseases early. After that, the KNN algorithm is applied to classify the type of disease based on the pattern that has been formed. By calculating the classification results using the KNN algorithm with a total of 94 data, as a manual calculation, 94 data were used, namely, 80% training data totaling 72, 20% test data totaling 22, obtaining 77.27% accuracy, 100% precision, 100% recall, and f1-score. 100%. By using the KNN algorithm to help classification, data in the form of attributes and labels is held out of training data and test data, where the labeling of training data must be determined at the beginning in the form of categorical information, namely Puccinia Polysora, Bipolar Maydis Syn and Helminthosporium Turcicum diseases.*

Keywords **:** *PCA, K-Nearest Neighbor, Corn plant diseases, Dimensionality reduction, Disease classification*

PENDAHULUAN

Komponen kehidupan yang paling penting di bumi ini adalah tumbuhan salah satu ciptaan Allah, tumbuh-tumbuhan dan buah-buahan banya di dijelaskan secara luar di dalam AL-Qur'an. Tumbuhan bermanfaat untuk bergagai hal,termasuk menyediakan oksigen untuk pernapasan, makanan, bahan bakar, obat-obatan.

Jagung ini merupakan salah satu jenis buah atau sayur yang bisa di konsumsi oleh masyarakat Indonesia. Satu pohon jagung dapat memproduksi sekitar 1 buah per 4 bulan sekalidalam keadaan ideal,yang merupakan tingkat produksi buah yang cukup tinggi untuk sebuah pohon. (Wahyuni, 2020).

Penyakit tanaman adalah suatu keadaan yang mengakibatkan timbulnya gejala dan menunjukkan bahwa jaringan dan sel tanaman tidak berfungsi dengan baik, di akibatkan gangguan terus-menerus oleh factor lingkungan. Demikian juga penyakiy yang menyerang tanaman jagung yaitu penyakit Hawar daun jagung (Helminthosporium turcicum), penyakit bercak pada daun (Bipolaris maydis syn). Karat daun (puccinia polysora) Informasi tentang penyakit jagung dan pengumpulan penyakit tergantung pada pengalaman. Saran dan blom otomatis yang setiap saat pasti menambah informasi baru tentang infeksi dan gangguan terbaru dengan menandai infeksi jagung yang baru-baru ini masuk ke dalam sistem,hingga petani dapat mengetahui dan memahami sekelompok infeksi. Berkumpulnya penyakit ini tergantung pada jenis penyakit dan keadaan bercak ke daun jagung, yang mencegah fotosentesis, menyebabkan pertumbuhan tanaman menjadi lambat. (purwandari, 2020)

Penyakit tanaman dapat diamati dari bentuk bercak pada daun dan tesktur daun merupakan ciri yang paling tepat untuk klasifikasi pada daun. (Furqan, 2021)

Namun identifikasi penyakit tanaman tidal mudah dilakukan, dikarenakan bentuk daun jagung yang beraneka ragam dengan demi kian, teknologi dapat membantu mendeteksi penyakit daun jagung. Perkembangan teknologi baru akan didominasi oleh sistem dan mesin mesin dengan kecerdasan buatan (machine intelligence).

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

Teknik pengenalan pola adalah paling penting dari mesin atau sistem cerdas yang digunakan untuk pemrosesan data dan pengambilan keputusan (putra, 2022)

Pengenalan pola pada daun dilakukan untuk mengklarisifikasi penyakit pada daun jagung dengan Metode Principal Component Analysis (PCA) dan K-Nearest Neighbor (KNN) menggunakan matlab. Proses pengenalan dapat ditangani secara efektif dengan metode Principal Component Analysis (PCA), karena dianggap mudah untuk mengekstrak informasi terkait dari seku pulan dat. Kumpulan data yang semula berkolerasi dapat diubah menjadi data yang tidak berkolerasi, dan jumlah Principal Component yang dihasilkan sama dengan jumlah data asli (Sugihartono, 2020).

Penelitian tentang Principal Component Analysis (PCA) dan K-Nearest Neighbor (KNN) telah banya digunakan dalam berbagai penelitian salah satunya yaitu penelitian yang berjudul "Implementasi Pengenalan Wajah Menggunakan PCA (Principal Component Analysis)". Penelitian ini menggunakan algoritma PCA sebagai metode ekstraksi ciri dalam proses identifikasi wajah, melalui beberapa tahapan yaitu Eigenface untuk dapat menghasilkan citra yang lebih baik. Kemudian proses pelatihan wajah yaitu mencari vector eigen, niai eigen dan rata-rata image yang diproyeksi ke dalam sub ruang PCA untuk menyederhanakan data citra yang tersimpan. Hasil nama pengguna ditentukan dengan membandingkan proyeksi PCA terkecil dicari menggunakan Nearest Neighbor. Presentase keberhasilan proses pengenalan wajah pada penelitian ini adalah 82,18% (Harjoko, 2021)

TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh Permadi Hidayat, Fitri Bimantoro dengan judul Deteksi Penyakit Hawar Daun Jagung Menggunakan Ekstrasi Fitur Glcm Dan Metode Artificial Neural Network Backpropagation (2023), Pada model arsitektur jaringan terbaik untuk deteksi penyakit pada daun jagung, dalam hal ini adalah menggunakan 1 hidden layer, learning rate sebesar 0.3, epoch 400, dan pembagian dataset sebesar 80:20. Model ini mencapai tingkat akurasi training sebesar 99.5%. Peningkatan akurasi model arsitektur ANN backpropagation setelah diuji ulang dengan penambahan jumlah dataset dari 3000 menjadi 3720 citra menghasilkan akurasi tertinggi sebesar 99.5%. Hal ini menunjukkan bahwa jumlah dataset juga memiliki pengaruh signifikan terhadap tingkat keakuratan model.

Penelitian yang dilakukan oleh Ayu sapitri, jangkung raharjo, syamsul rizal dengan judul Identifikasi Penyakit Jagung Dengan Menerapkan Metode Gray Level CoOccurrence Matrix (GLCM) Dan Support Vector Machine (SVM) Melalui Citra Daun (2022), Sistem yang dirancang pada penelitian ini menggunakan metode GLCM dan klasifikasi SVM dan telah melalui tahap pengujian dan analisis. Jika dilihat dari nilai akurasi metode yang digunakan cukup efisien jika di aplikasikan untuk sistem klasifikasi penyakit pada daun jagung. Adapun beberapa kesimpulan yang idapat melaluui hasil pengujian sistem diantarnaya adalah: 1. Penelitian ini telah melewati proses ektrasi ciri menggunakan metode GLCM serta diklasifikasikan dengan metode SVM untuk mendapatkan nilai akurasi yang terbaik. Citra yang digunakan pada penelitian ini berupa gambar daun jagung yang terdiri dari 4 kelas penyakit daun diantarnya adalah daun sehat, daun hawar, karat daun, serta bercak daun.

Penelitian yang dilakukan oleh Setya Putra Adenugraha Veri Arinal, Dadang Iskandar Mulyana dengan judul Klasifikasi kematangan Buah Pisang Ambon Menggunakan Metode KNN dan PCA Berdasarkan citra RGB dan HSV (2022), Penelitian ini mengembangkan sistem untuk mengklasifikasikan tingkat kematangan pisang ambon menggunakan fitur warna RGB dan HSV. Pada penelitian ini digunakan 41 data yang terbagi menjadi 30 data latih dan 11 data uji, dan klasifikasi dengan tiga kelas yaitu mentah, matang, dan sangat matang. Metode KNN digunakan untuk mengklasifikasikan data dengan menentukan jarak tetangga terdekat dengan nilai K=5. Hasil penelitian menenjukkan akurasi sebesar 99,9%.

Principal Componen Analysis (PCA)

Analisis komponen utama (PCA) adalah metode statistik yang dapat menguraikan matriks data menjadi matriks vektor yang disebut komponen utama. Meskipun analisis komponen utama pada dasarnya adalah teknik reduksi dimensi juga dapat digunakan untuk berbagai tujuan, seperti memeriksa kumpulan data untuk menemukan outlier. PCA (Principal Component Analysis) yang dikembangkan pada dapat digunakan untuk mendeteksi outlier PCA (Principal Component Analysis) merupakan teori dan metode yang bertujuan untuk mendeteksi outlier, terlebih dahulu mencocokkan sebagian besar data kemudian menandai titik-titik yang menyimpang.Tugas akhir ini memungkinkan untuk mengubah matriks menjadi matriks ortogonal. Vektor tersebut dapat didekomposisi sedemikian rupa sehingga menunjukkan vektor eigen yang terkait dengan nilai eigen . Vektor eigen dengan nilai eigen tinggi sebesar dapat menangkap sebagian besar varians dalam data Hyperplane berdimensi rendah yang dibuat oleh vektor eigen k kemudian dapat menangkap sebagian besar varians dalam data. Namun, outlier berbeda dari titik data normal yang lebih jelas di hyperplane yang dibangun berdasarkan vektor eigen dan nilai eigennya yang kecil. Untuk di antaranya, outlier dapat ditentukan sebagai jumlah dari proyeksi jarak sampel untuk semua vektor eigen [7].

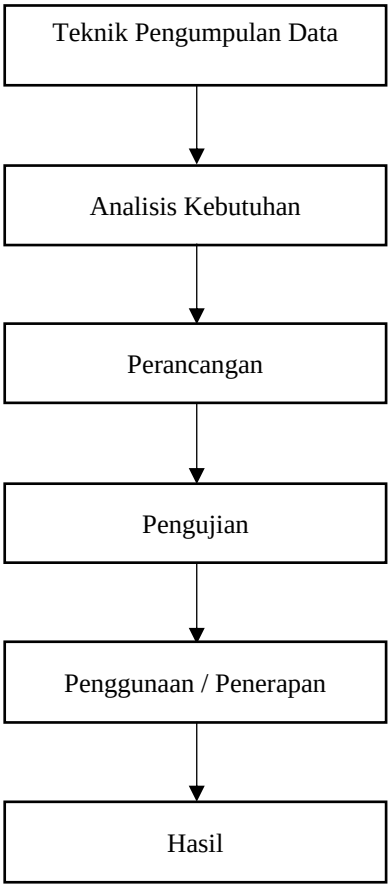
Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

K-Nearet Neighbor (KNN)

Algoritma KNN mengklasifikasikan objek berdasarkan kelas mayoritas dari objek terdekat di lingkungan.RephraseMetode ini memiliki algoritma yang sederhana dan cocok digunakan dengan dataset kecil.Namun algoritma ini sangat bergantung pada nilai k .Pemilihan nilai k mempunyai pengaruh yang signifikan terhadap hasil klasifikasi .Umumnya jika ingin mencari k benda terdekat dapat dilakukan dengan melihat jarak antar benda. K-Nearest Neighbors (KNN) merupakan metode yang menggunakan algoritma terawasi dimana hasil query instance baru diklasifikasikan berdasarkan sebagian besar kategori yang ada di KNN.Tujuan dari algoritma ANN adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan pola pelatihannya.Hasil sampel pengujian baru diklasifikasikan berdasarkan sebagian besar kategori yang ada di ANN K- Nearest Neighbor merupakan metode klasifikasi objek berdasarkan tingkat kemiripan atau kesamaan pada sampel data training (data latih) melalui klasifikasi jarak maupun jumlah tetangga terdekat. Tetangga terdekat ditentukan oleh fungsi matrik, yaitu berdasarkan nilai k yang didefinisikan dari titik pada sampel data.[8].

METODE

Proses kerja penelitian terdiri dari beberapa tahapan yang menggambarkan seluruh tahapan kegiatan yang dilakukan selama penelitian untuk mencapai tujuan yang telah ditentukan. Di bawah ini adalah langkah-langkah yang harus dilakukan.



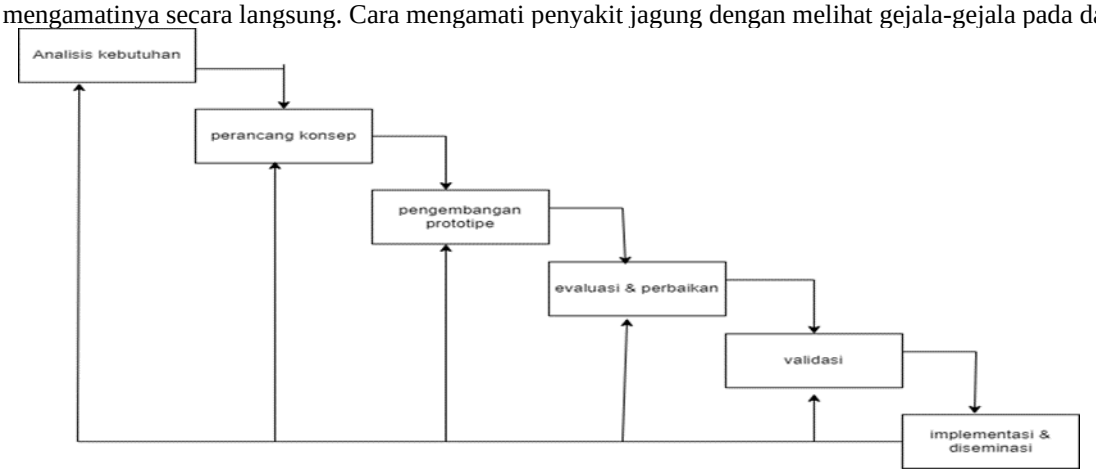
Gambar 1. Tahapan Kerja Penelitian

Teknik Pengumpulan data berikut adalah teknik pengumpulan data:

1. Penelitian perpustakan
   Dalam penelitian kepustakaan ini, penulis mencari jurnal dan e-book untuk mendapatkan wawasan dan mengunpulkan referensi teori dasr dari berbagaiartikel dan jurnal di internet.
2. Studi Literatur
   Studi literature melibatkan berbagai kegiatan seperti mengumpulkan ninliografi, membaca dan mencatat, secara menganalisis data penelitian atau mecari referensi teoritis tentang suatu kasus atau masalah yang dibahas dalam tugas akhir ini
3. Observasi

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

Observasi adalah pengumpulan data yang dilakukan dengan cara meninjau daun tanaman jagung untuk mengamatinya secara langsung. Cara mengamati penyakit jagung dengan melihat gejala-gejala pada daun.



Gambar 2. Tahapan Diagram R dan D

1. Analisis Kebutuhan
   Dilakukan analisis kebutuhan untuk menentukan kebutuhan sistem yang dibangun, seperti jenis data yang diperlukan, tingkat akurasi yang diharapkan, dan kendala-kendala yang mungkin dihadapi dalam pengenalan pola penyakit
2. Perancang Konsep
   Konsep sistem pengenalan pola penyakit tanaman jagung dirancang berdasarkan hasil analisis kebutuhan. Ini mencakup pemilihan metode PCA dan KNN sebagai teknik utama untuk ekstraksi fitur dan klasifikasi pola penyakit.
3. Pengembangan Prototipe
   Prototipe sistem dikembangkan untuk menguji konsep yang dirancang. Data penyakit tanaman jagung digunakan untuk melatih model PCA dan KNN, serta untuk menguji performanya.
4. Evaluasi & Perbaiki
   Prototipe dievaluasi untuk mengukur kinerjanya dalam mengenali pola penyakit. Jika ditemukan kelemahan atau area peningkatan, dilakukan perbaikan dan penyesuaian pada konsep dan implementasi sistem.
5. Validasi
   Validasi dilakukan dengan menguji prototipe pada dataset yang berbeda atau menguji keandalan sistem dalam kondisi yang berbeda. Ini memastikan bahwa sistem dapat digunakan secara luas dan dapat diandalkan.
6. Implementasi & Diseminasi
   Setelah prototipe diverifikasi dan divalidasi, dilakukan implementasi sistem untuk digunakan dalam pemantauan penyakit tanaman jagung di lapangan. Hasil penelitian dan pengembangan juga dapat didiseminasi melalui publikasi ilmiah dan presentasi di konferensi.

Dengan menggunakan pendekatan R&D, penelitian ini akan memastikan bahwa sistem pengenalan pola penyakit tanaman jagung yang dikembangkan menggunakan metode PCA dan KNN dapat memenuhi kebutuhan praktis dan ilmiah yang diharapkan.

HASIL DAN PEMBAHASAN

Dalam menganalisa dan merancang model yang baik, dibutuhkan data dan informasi yang tepat dan bersesuaian dengan kebutuhan model. Berikut merupakan total data set penyakit daun terhadap terhadap kebun jagung buk elvi yang didapat dari hasil riset berjumlah 94, kelas yang ditentukan disini berupa: Helminthosporium Turcicum (penyakit hawar), Bipolaris Maydis Syn(penyakit bercak) dan Puccinia Polysora (penyakit karat) kelas yang didapat diperoleh langsung dari tempat riset, kelas penyakit daun digunakan kedepannya hanya untuk pengujian akurasi, untuk data testing tidak diharuskan memiliki kelas penyakit daun. Rancangan antarmuka dalam coding di Google Colab merujuk pada desain dan implementasi elemen yang memungkinkan pengguna berinteraksi dengan notebook secara lebih efisien dan intuitif. Di Google Colab, antarmuka dalam penggunaan PCA dan K-NN dalam klasifikasi penyakit daun jagung umumnya

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

Tabel 1.Total Data Set Penyakit Daun Jagung
Sumber : (kebun jagung buk Elvi)

No	Warna Daun	Tekstur Daun	Suhu (°C)	Kelembapan (%)	Intensitas Cahaya	Bentuk Bercak	Tepi Daun	Penyakit
1	Hijau	Halus	24,68	73,68	Sedang	Bulat	Bergerigi	Puccinia polysora
2	Kuning	Kasar	26,38	69,06	Sedang	Bulat	Bergerigi	Bipolaris maydis (Syn)
3	Hijau	Halus	33,28	69,52	Tinggi	Lonjong	Bergerigi	Bipolaris maydis (Syn)
4	Kuning	Kasar	30,2	68,34	Rendah	Lonjong	Bergerigi	Helminthosporium turcicum
5	Kuning	Kasar	26,84	70,98	Sedang	Bulat	Rata	Helminthosporium turcicum
6	Hijau	Kasar	28,12	75,63	Tinggi	Bulat	Melengkung	Bipolaris maydis (Syn)
7	Coklat	Halus	20,43	80,94	Rendah	Tidak Beraturan	Rata	Puccinia polysora
8	Kuning	Halus	33,5	88,76	Tinggi	Lonjong	Melengkung	Helminthosporium turcicum
9	Hijau	Kasar	32,67	78,14	Sedang	Bulat	Rata	Puccinia polysora
10	Hijau	Halus	29,84	81,22	Tinggi	Bulat	Rata	Helminthosporium turcicum
11	Kuning	Kasar	25,97	82,11	Sedang	Lonjong	Melengkung	Bipolaris maydis (Syn)
12	Coklat	Halus	34,45	77,38	Rendah	Tidak Beraturan	Bergerigi	Puccinia polysora
13	Hijau	Kasar	30,66	68,98	Tinggi	Bulat	Rata	Helminthosporium turcicum
14	Kuning	Halus	27,34	74,56	Sedang	Lonjong	Melengkung	Bipolaris maydis (Syn)
15	Hijau	Kasar	28,57	65,43	Rendah	Bulat	Bergerigi	Puccinia polysora
16	Coklat	Halus	22,48	71,15	Tinggi	Tidak Beraturan	Rata	Helminthosporium turcicum
17	Kuning	Kasar	31,72	73,34	Sedang	Lonjong	Melengkung	Bipolaris maydis (Syn)
18	Hijau	Halus	33,89	78,23	Rendah	Bulat	Bergerigi	Puccinia polysora
19	Kuning	Halus	28,73	72,45	Sedang	Lonjong	Melengkung	Bipolaris maydis (Syn)
20	Hijau	Kasar	32,14	67,98	Rendah	Bulat	Bergerigi	Puccinia polysora

Data yang digunakan yaitu, kualitatif dan kuantitatif atau kategorik dan numerik. Data kategorik adalah jenis data yang mengandung nilai-nilai yang bersifat kategori atau label, bukan angka, data numerik adalah jenis data yang mengandung nilai-nilai yang bersifat angka dapat diukur dan dioperasikan secara matematis.

Rancangan antarmuka dalam coding di Google Colab merujuk pada desain dan implementasi elemen yang memungkinkan pengguna berinteraksi dengan notebook secara lebih efisien dan intuitif. Di Google Colab, antarmuka dalam penggunaan PCA dan K-NN dalam klasifikasi penyakit daun jagung umumnya berupa :

1. Kode ini mengimpor pustaka dan modul yang digunakan dalam analisis data dan machine learning: pandas untuk manipulasi data, numpy untuk komputasi numerik, dan matplotlib untuk visualisasi. Selain itu, scikit-learn digunakan untuk preprocessing data dan evaluasi model, termasuk standarisasi, pengkodean label, dan metrik evaluasi seperti akurasi dan matriks kebingungan. Terakhir, modul files dari Google Colab memfasilitasi upload dan download file di lingkungan Colab.

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from google.colab import files
```

Gambar 3. Pustaka yang Digunakan

2. Dalam pembacaan data disini digunakan file upload menggunakan file excel, kode ini digunakan untuk meng-upload dan membaca file Excel di Google Colab.

```
# Upload the Excel file
uploaded = files.upload()

# Read the Excel file
file_name = list(uploaded.keys())[0]
df = pd.read_excel(file_name)
```

Gambar 4. Metode Dalam Pembacaan Data

3. Kode ini menampilkan beberapa baris pertama dari DataFrame yang baru dibaca untuk memeriksa isinya. Selanjutnya, kode mengkodekan variabel kategorikal dengan menggunakan LabelEncoder, mengonversi setiap kolom bertipe objek menjadi nilai numerik. Hasil pengkodean disimpan dalam DataFrame yang sama. Terakhir, kode memisahkan data menjadi fitur (X) dan variabel target (y), di mana X berisi semua kolom kecuali kolom "Penyakit" dan y berisi kolom "Penyakit" sebagai target prediksi.

```
# Display the first few rows of the dataframe
print("Original DataFrame:")
print(df.head())

# Encode categorical variables
label_encoders = {}
for column in df.select_dtypes(include=["object"]).columns:
    label_encoders[column] = LabelEncoder()
    df[column] = label_encoders[column].fit_transform(df[column])

# Separate features and target variable
X = df.drop("Penyakit", axis=1)
y = df["Penyakit"]
```

Gambar 5. *Encoding Categorical Attribut*

4. Kode ini melakukan normalisasi dan perhitungan PCA secara manual. Pertama, data fitur X dinormalisasi menggunakan StandardScaler, menghasilkan X_scaled dengan fitur yang memiliki rata-rata 0 dan deviasi standar 1. Selanjutnya, untuk PCA manual, kode menghitung matriks kovarians dari data yang dinormalisasi dengan np.cov(X_scaled.T), di mana .T mentransposisikan data agar kovarians dihitung antar fitur. Matriks kovarians ini kemudian ditampilkan untuk langkah berikutnya dalam analisis PCA.

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

```
# Normalize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Manual PCA calculation
print("\nStep-by-Step PCA Calculation:")

# 1. Calculate the covariance matrix
cov_matrix = np.cov(X_scaled.T)
print("Covariance Matrix:")
print(cov_matrix)
```

Gambar 6. Normalisasi PCA

5. Kode ini menghitung nilai dan vektor eigen dari matriks kovarians menggunakan np.linalg.eig(), yang diperlukan untuk langkah reduksi dimensi dalam PCA.

```
# 2. Compute eigenvalues and eigenvectors
eigen_values, eigen_vectors = np.linalg.eig(cov_matrix)
print("\nEigenvalues:")
print(eigen_values)
print("\nEigenvectors:")
print(eigen_vectors)
```

Gambar 7. Hitung *Eigen Value* dan *Eigen Vector*

6. Kode ini mengurutkan nilai dan vektor eigen dari yang terbesar ke yang terkecil. np.argsort(eigen_values)[::-1] menghasilkan urutan indeks, yang digunakan untuk menyusun ulang eigen_vectors dan eigen_values. Hasilnya, nilai dan vektor eigen yang telah diurutkan dicetak.

```
# 3. Sort eigenvalues and eigenvectors
sorted_index = np.argsort(eigen_values)[::-1]
sorted_eigenvectors = eigen_vectors[:, sorted_index]
sorted_eigenvalues = eigen_values[sorted_index]
print("\nSorted Eigenvalues:")
print(sorted_eigenvalues)
print("\nSorted Eigenvectors:")
print(sorted_eigenvectors)
```

Gambar 8. Hitung *Eigen Value* dan *Eigen Vector*

7. Kode ini memilih n_components vektor eigen teratas untuk reduksi dimensi, mengakses kolom pertama dari sorted_eigenvectors sesuai dengan jumlah komponen yang diinginkan (n_components). Hasilnya, vektor eigen yang dipilih ditampilkan.

```
# 4. Select the top n_components
n_components = 2
eigenvector_subset = sorted_eigenvectors[:, 0:n_components]
print("\nTop {} Eigenvectors:".format(n_components))
print(eigenvector_subset)
```

Gambar 9. *Vector Eigen* Teratas

8. Kode ini mentransformasi data yang dinormalisasi ke dalam ruang dimensi yang direduksi dengan mengalikan X_scaled dengan eigenvector_subset. Hasilnya, data yang telah direduksi PCA ditampilkan sebagai DataFrame dengan kolom "PC1" dan "PC2".

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

```
# 5. Transform the data
X_pca_manual = np.dot(X_scaled, eigenvector_subset)
print("\nPCA-Reduced Data (Manual Calculation):")
print(pd.DataFrame(X_pca_manual, columns=["PC1", "PC2"]))
```

Gambar 10. Transformasi Data

9. Kode ini membagi data yang telah direduksi PCA menjadi data pelatihan dan pengujian. train_indices mencakup indeks dari 0 hingga 79 untuk pelatihan, sedangkan test_indices mencakup sisanya untuk pengujian. X_train dan X_test adalah fitur pelatihan dan pengujian, sementara y_train dan y_test adalah label yang sesuai.

```
# Specify custom training and testing data indices
train_indices = list(range(0, 80))  # First 80 samples for training
test_indices = list(range(80, len(df)))  # Remaining samples for testing

X_train, X_test = X_pca_manual[train_indices], X_pca_manual[test_indices]
y_train, y_test = y[train_indices], y[test_indices]
```

Gambar 11. Pembagian Data Latih dan Data Uji

10. Kode ini mendefinisikan fungsi euclidean_distance untuk menghitung jarak Euclidean antara dua vektor a dan b. Fungsi ini menghitung akar kuadrat dari jumlah kuadrat perbedaan elemen-elemen yang bersesuaian dari a & b.

```
# Manual KNN calculation
def euclidean_distance(a, b):
    return np.sqrt(np.sum((a - b) ** 2))
```

Gambar 12. Pembagian Data Latih dan Data Uji

11. Fungsi knn_predict melakukan prediksi menggunakan algoritma K-Nearest Neighbors (KNN). Untuk setiap titik uji (X_test), fungsi ini :
    a. Menghitung jarak Euclidean ke semua titik pelatihan (X_train).
    b. Menentukan k tetangga terdekat berdasarkan jarak terkecil.
    c. Mengambil label dari tetangga terdekat dan menentukan label yang paling sering muncul.
    d. Menyimpan prediksi dan mencetak detail jarak, indeks, dan label tetangga terdekat untuk setiap titik uji.

    Fungsi ini mengembalikan daftar prediksi untuk titik uji.

```
def knn_predict(X_train, y_train, X_test, k=3):
    predictions = []
    for test_index, test_point in enumerate(X_test):
        distances = [euclidean_distance(test_point, train_point) for train_point in X_train]
        k_indices = np.argsort(distances)[:k]
        k_nearest_labels = [y_train[i] for i in k_indices]
        most_common = np.bincount(k_nearest_labels).argmax()
        predictions.append(most_common)
        # Print detailed distance and neighbor information
        print("\nTest Point {}:".format(test_index + 1))
        print("Distances to training points:")
        for i, dist in enumerate(distances):
            print("Distance to training point {}: {}".format(i + 1, dist))
        print("Indices of k-nearest neighbors: {}".format(k_indices + 1))
        print("Labels of k-nearest neighbors: {}".format(k_nearest_labels))
        print("Predicted label: {}".format(most_common))
    return predictions
```

Gambar 13. Fungsi Prediksi K-NN

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

12. Kode ini menjalankan fungsi knn_predict untuk mendapatkan prediksi pada data uji (X_test) dengan k=3. Kemudian, accuracy_score menghitung akurasi prediksi dibandingkan dengan label sebenarnya (y_test). Akurasi dikonversi ke persentase dengan mengalikan hasilnya dengan 100.

```
y_pred_manual = knn_predict(X_train, y_train, X_test, k=3)
accuracy_manual = accuracy_score(y_test, y_pred_manual)
accuracy_new = accuracy_manual * 100
```

Gambar 14. Fungsi Akurasi

13. Kode ini mendekode label numerik kembali ke format asli menggunakan inverse_transform dari LabelEncoder untuk memudahkan pembacaan. Kemudian, kode mencetak perbandingan antara prediksi dan nilai aktual untuk melihat hasil manual KNN. Akurasi manual ditampilkan sebagai persentase. Terakhir, classification_report digunakan untuk menghitung dan menampilkan metrik evaluasi seperti precision, recall, dan F1-score untuk prediksi.

```
# Decode labels for better readability
y_test_decoded = label_encoders["Penyakit"].inverse_transform(y_test)
y_pred_manual_decoded = label_encoders["Penyakit"].inverse_transform(y_pred_manual)
y_train_decoded = label_encoders["Penyakit"].inverse_transform(y_train)

# Display predictions and actual values (manual calculation)
print("\nPredictions vs Actual (Manual Calculation):")
for pred, actual in zip(y_pred_manual_decoded, y_test_decoded):
    print(f"Predicted: {pred}, Actual: {actual}")

print("Manual KNN Accuracy:", accuracy_new, "%")

# Calculate and display precision, recall, and F1 score
report = classification_report(y_test_decoded, y_pred_manual_decoded)
print("\nClassification Report:")
print(report)
```

Gambar 15. Menampilkan Hasil Akhir Klasifikasi K-NN

14. Kode ini membuat visualisasi data yang telah direduksi PCA :
    a. Plot Data Pelatihan: Menampilkan data pelatihan (X_train) dengan warna yang berbeda untuk setiap label asli pada subplot pertama.
    b. Plot Data Uji dengan Prediksi: Menampilkan data uji (X_test) dengan dua penanda: satu untuk label aktual dan satu lagi untuk prediksi, pada subplot kedua.

Visualisasi ini membantu dalam memahami bagaimana data dipisahkan dan bagaimana prediksi dibandingkan dengan label sebenarnya.

```
# Visualize the PCA-reduced data
plt.figure(figsize=(14, 7))

# Plot the training data
plt.subplot(1, 2, 1)
for label in np.unique(y_train_decoded):
    plt.scatter(X_train[y_train_decoded == label, 0], X_train[y_train_decoded == label, 1], label=label)
plt.title('PCA-Reduced Training Data')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()

# Plot the test data with predictions
plt.subplot(1, 2, 2)
for label in np.unique(y_test_decoded):
    plt.scatter(X_test[y_test_decoded == label, 0], X_test[y_test_decoded == label, 1], label=f'Actual {label}')
for label in np.unique(y_pred_manual_decoded):
    plt.scatter(X_test[y_pred_manual_decoded == label, 0], X_test[y_pred_manual_decoded == label, 1], marker='x', label=f'Predicted {label}')
plt.title('PCA-Reduced Test Data with Predictions')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()

plt.tight_layout()
plt.show()
```

Gambar 16. Menampilkan Hasil Plot

Jurnal Informatika Teknologi dan Sains (JINTEKS)
E-ISSN : 2686-3359
Submit July 30, 2024 | Revised Agustus 05, 2024 |
Accepted Agustus 06, 2024
Hal. 659-668
Vol. 6; No. 3
Agustus 2024

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian klasifikasi dengan KNN dalam menentukan penyakit daun jagung, penulis mengambil kesimpulan sebagai berikut, Dengan menghitung hasil klasifikasi menggunakan algoritma KNN dengan total data 94, sebagai perhitungan manual digunakan 94 data yaitu, data latih 80% berjumlah 72, data uji 20% berjumlah 22 diperoleh acuracy 77.27%, precision 100%, recall 100%, dan f1-score 100%. Dengan menggunakan algoritma KNN membantu klasifikasi, data berupa attribut dan label dilakukan hold out data latih dan data uji, dimana pelabelan data latih harus ditentukan diawal berupa keterangan kategorik yaitu penyakit Puccinia Polysora, Bipolar Maydis Syn dan Helminthosporium Turcicum.

Berdasarkan uraian dan pembahasan pada bab-bab sebelumnya, maka terdapat beberapa saran sebagai berikut, Model dapat dikembangkan agar menggunakan metode lain untuk mendapatkan hasil yang lebih maksimal. Model perlu ditambah data latih agar mendapatkan hasil maksimal dengan perhitungan untuk klasifikasi.

DAFTAR PUSTAKA

[1] B. Raden, K. Bogor, and J. Barat, "SIKAMA : Jurnal Pengabdian Kepada Masyarakat," vol. 1, no. 2, pp. 70–78, 2023
[2] E. Solihin, R. Sudirja, H. Maulana, and N. N. Kamaluddin, "Respon Pertumbuhan dan Hasil Tanaman Jagung (Zea Mays L .) Terhadap Pemberian Pupuk Majemuk P dan K," vol. 3, no. 2, pp. 20–23, 2023
[3] L. Di, S. Menang, K. Gelumbang, M. Enim, and S. Selatan, "Planta Simbiosa," vol. 6, no. April, pp. 1–19, 2024
[4] M. E. Lestari, I. Asror, and I. L. Sardi, "Penerapan PCA (Principal Component Analysis) pada Deteksi Outlier untuk Data Text," *e-Proceeding Eng.*, vol. 10, no. 3, pp. 3549–3555, 2023
[5] N. P. Indriani, H. K. Mustafa, and R. Z. Islami, "Introduksi Tanaman Penghasil Jagung Semi (Zea mays) dan Hijauan Pakan dengan Berbagai Varietas dan Umur Panen di Desa Cileles Kabupaten Sumedang," *Media Kontak Tani Ternak*, vol. 4, no. 2, pp. 50–55, 2022
[6] Pola, P., Tumpangsari, T., Zea, J., Padi, V., Sunaryo, Y., Respati, L. D., Arnanto, D., Batikan, J., Umbulharjo, K., Yogyakarta, K., & Yogyakarta, D. I. (2024). *(Oryza sativa) TERHADAP HASIL DI LAHAN KERING GUNUNGKIDUL THE EFFECT OF INTERCROPPING OF CORN (Zea mays) WITH 3 VARIETIES OF RICE (Oryza sativa) ON DRY LAND IN GUNUNGKIDUL PENDAHULUAN Jagung (Zea mays L .)*
[7] Tangkelayuk, A., & Mailoa, E. (2022). Klasifikasi Kualitas Air Menggunakan Metode KNN , Naïve Bayes Dan Decision Tree. 9(2), 1109–1119.
[8] S. Margaretta, I. Arwani, and D. E. Ratnawati, "Implementasi Algoritma K-Nearest Neighbor Pada Database Menggunakan Bahasa SQL," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 7, pp. 2043–2052, 2020
[9] Solihin, E., Sudirja, R., Maulana, H., & Kamaluddin, N. N. (2023). *Respon Pertumbuhan dan Hasil Tanaman Jagung*
[10] V. Nomor, "PERTUMBUHAN DAN HASIL JAGUNG MANIS (Zea mays SACCHARATA STURT) VARIETAS PARAGON AKIBAT PERLAKUAN JARAK TANAM DAN JUMLAH BENIH," vol. 4, pp. 1–10, 2022
[11] Khairiyah, A. (2020). Klasifikasi Jenis Buah Jambu Berdasarkan Daun Menggunakan Metode Principal Component Analysis. 18–18.
[12] Kus Indrani Listyoningrum, Danise Yunaini Fenida, & Nurhasan Hamidi. (2023). Inovasi Berkelanjutan dalam Bisnis: Manfaatkan Flowchart untuk Mengoptimalkan Nilai Limbah Perusahaan. Jurnal Informasi Pengabdian Masyarakat, 1(4), 100–112. https://doi.org/10.47861/jipm-nalanda.v1i4.552
[13] Lestari, M. E., Asror, I., & Sardi, I. L. (2023). Penerapan PCA (Principal Component Analysis) pada Deteksi Outlier untuk Data Text. E-Proceeding of Engineering, 10(3), 3549–3555. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/20626
[14] Margaretta, S., Arwani, I., & Ratnawati, D. E. (2020). Implementasi Algoritma K-Nearest Neighbor Pada Database Menggunakan Bahasa SQL. Pengembangan Teknologi Informasi Dan Ilmu Komputer, 4(7), 2043–2052. http://j-ptiik.ub.ac.id
[15] Indriani, N. P., Mustafa, H. K., & Islami, R. Z. (2022). Introduksi Tanaman Penghasil Jagung Semi (Zea mays) dan Hijauan Pakan dengan Berbagai Varietas dan Umur Panen di Desa Cileles Kabupaten Sumedang. *Media Kontak Tani Ternak*, *4*(2), 50–55.