

## IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBORS PADA APLIKASI WEB UNTUK KLASIFIKASI BUNGA IRIS

Jihad Akbar<sup>1</sup>.

**ABSTRAK:** Identifikasi spesies bunga Iris secara manual seringkali sulit dilakukan karena kemiripan fitur morfologis antar spesiesnya. Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem klasifikasi otomatis berbasis web untuk memprediksi spesies bunga Iris. Metode yang digunakan adalah algoritma K-Nearest Neighbors (KNN) yang diimplementasikan menggunakan Python dengan pustaka Scikit-Learn. Dataset yang digunakan adalah dataset Iris standar yang terdiri dari 150 sampel dengan 4 fitur numerik. Model KNN dievaluasi menggunakan metrik akurasi dan confusion matrix, kemudian diintegrasikan ke dalam aplikasi web yang dibangun dengan framework Flask dan antarmuka Bootstrap. Hasil penelitian menunjukkan bahwa model KNN mencapai akurasi sebesar 100% pada data uji dengan nilai  $k=3$ . Aplikasi web yang dikembangkan berhasil mengimplementasikan model tersebut, memungkinkan pengguna untuk melakukan prediksi spesies secara interaktif dan akurat. Simpulan dari penelitian ini adalah algoritma KNN sangat efektif untuk kasus klasifikasi Iris dan implementasinya pada platform web memberikan solusi praktis yang mudah diakses.

**Kata kunci :** Klasifikasi, K-Nearest Neighbors, Bunga Iris, Flask, Machine Learning.

**ABSTRACT:** Manual identification of Iris flower species is often difficult due to the similarity of morphological features between species. This research aims to design and build a web-based automatic classification system to predict Iris flower species. The method used is the K-Nearest Neighbors (KNN) algorithm, implemented using Python with the Scikit-Learn library. The dataset used is the standard Iris dataset, consisting of 150 samples with 4 numerical features. The KNN model was evaluated using accuracy and a confusion matrix, then integrated into a web application built with the Flask framework and a Bootstrap interface. The research results show that the KNN model achieved an accuracy of 100% on the test data with a  $k$ -value of 3. The developed web application successfully implemented the model, allowing users to perform species prediction interactively and accurately. The conclusion of this research is that the KNN algorithm is highly effective for the Iris classification case, and its implementation on a web platform provides a practical and easily accessible solution.

**Keywords:** Classification, K-Nearest Neighbors, Iris Flower, Flask, Machine Learning.

## PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (AI) dan *machine learning* telah memberikan dampak signifikan di berbagai bidang, termasuk botani dan biologi. Salah satu tantangan dalam botani adalah identifikasi spesies tanaman secara cepat dan akurat, yang seringkali membutuhkan keahlian khusus. Bunga Iris, dengan tiga spesies utamanya yaitu *Iris setosa*, *Iris versicolor*, dan *Iris virginica*, merupakan contoh klasik di mana identifikasi bisa menjadi sulit karena beberapa spesies memiliki karakteristik visual yang tumpang tindih.

K-Nearest Neighbors (KNN) adalah salah satu algoritma *supervised learning* yang fundamental dan kuat untuk masalah klasifikasi. Cara kerjanya yang intuitif, yaitu mengklasifikasikan data baru berdasarkan mayoritas kelas dari ‘tetangga’ terdekatnya, membuat algoritma ini mudah dipahami dan diimplementasikan. Konsep "kedekatan" diukur menggunakan metrik jarak, seperti Jarak Euclidean, yang menghitung jarak garis lurus antara dua titik dalam ruang fitur.

Namun, model *machine learning* yang andal akan lebih bermanfaat jika dapat diakses dengan mudah oleh pengguna awam. Oleh karena itu, penelitian ini bertujuan untuk tidak hanya membangun model klasifikasi yang akurat menggunakan KNN, tetapi juga mengimplementasikannya dalam sebuah aplikasi berbasis web. Dengan menggunakan *framework* Flask, aplikasi ini akan menyediakan antarmuka yang ramah pengguna, memungkinkan siapa saja untuk memasukkan data fitur bunga Iris dan mendapatkan hasil prediksi spesies secara instan. Penelitian ini akan menjadi studi kasus lengkap dari pengembangan model hingga deployment aplikasi fungsional.

## METODA

**1. Dataset** Penelitian ini menggunakan dataset Iris yang merupakan dataset standar untuk masalah klasifikasi. Dataset ini diperoleh dari pustaka scikit-learn. Dataset ini terdiri dari:

- Jumlah Sampel: 150 data
- Jumlah Kelas: 3 spesies (Setosa, Versicolor, Virginica)
- Jumlah Fitur: 4 fitur numerik, yaitu:
  - Panjang Kelopak (Sepal Length) dalam cm
  - Lebar Kelopak (Sepal Width) dalam cm
  - Panjang Mahkota (Petal Length) dalam cm
  - Lebar Mahkota (Petal Width) dalam cm

**2. Alat dan Pustaka** Proses pengembangan model dan aplikasi menggunakan bahasa pemrograman Python 3 dengan beberapa pustaka utama, antara lain:

- **Scikit-learn:** Untuk implementasi algoritma KNN, pra-pemrosesan data, dan evaluasi model.
- **Pandas:** Untuk manipulasi dan analisis data.
- **Matplotlib & Seaborn:** Untuk visualisasi data.
- **Joblib:** Untuk menyimpan model yang telah dilatih.
- **Flask:** Sebagai *framework backend* untuk membangun aplikasi web.

**3. Tahapan Penelitian** Alur penelitian ini dibagi menjadi beberapa tahap berikut:

- **Pra-pemrosesan Data:** Data dibagi menjadi dua set, yaitu data latih (80%) dan data uji (20%) menggunakan fungsi `train_test_split` dari Scikit-learn untuk memastikan evaluasi model yang objektif.

- **Parameter Tuning:** Dilakukan pencarian nilai k (jumlah tetangga) yang optimal dengan menguji rentang nilai k dari 1 hingga 30 dan memilih nilai yang memberikan akurasi tertinggi pada data uji.
- **Pelatihan Model:** Model KNN final dilatih menggunakan data latih dengan nilai k optimal yang telah ditemukan.
- **Evaluasi Model:** Kinerja model dievaluasi menggunakan metrik Akurasi, *Classification Report* (yang mencakup presisi, recall, f1-score), dan *Confusion Matrix*.
- **Pengembangan Aplikasi Web:** Model yang telah dilatih disimpan menggunakan joblib. Sebuah aplikasi web dibangun menggunakan Flask, di mana backend menangani logika pemrosesan input dan prediksi, sedangkan frontend yang dibangun dengan HTML dan Bootstrap menyediakan antarmuka bagi pengguna.

## HASIL DAN PEMBAHASAN

### 1. Hasil Pelatihan dan Evaluasi Model

Dari proses *parameter tuning*, ditemukan bahwa model KNN mencapai akurasi maksimal pada beberapa nilai k, salah satunya adalah k=3. Dengan menggunakan k=3, model dilatih ulang dan dievaluasi pada data uji. Hasil evaluasi menunjukkan kinerja yang sempurna.

Akurasi model pada data uji adalah **100%**. Artinya, model berhasil

memprediksi dengan benar seluruh 30 sampel data pada set pengujian.

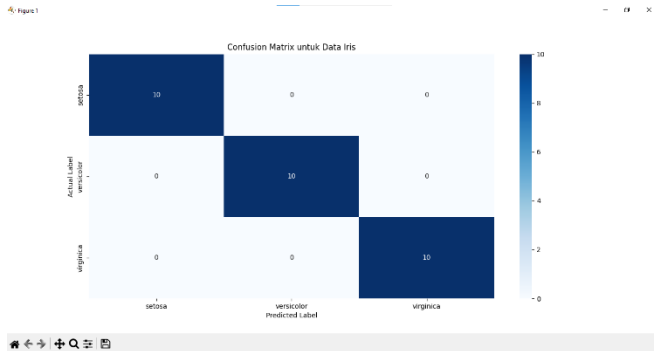
Laporan klasifikasi detailnya adalah seperti pada gambar 1 berikut:

| Laporan Klasifikasi: |           |        |          |         |
|----------------------|-----------|--------|----------|---------|
|                      | precision | recall | f1-score | support |
| setosa               | 1.00      | 1.00   | 1.00     | 10      |
| versicolor           | 1.00      | 1.00   | 1.00     | 10      |
| virginica            | 1.00      | 1.00   | 1.00     | 10      |
| accuracy             |           |        | 1.00     | 30      |
| macro avg            | 1.00      | 1.00   | 1.00     | 30      |
| weighted avg         | 1.00      | 1.00   | 1.00     | 30      |

Gambar 1 Laporan Klasifikasi

Hasil di atas menunjukkan bahwa untuk ketiga kelas (*setosa*, *versicolor*, dan *virginica*), nilai presisi, recall, dan f1-score semuanya adalah 1.00. Ini mengindikasikan tidak ada satupun kesalahan klasifikasi yang dibuat oleh model pada data uji.

*Confusion Matrix* juga mengonfirmasi hasil ini, di mana semua nilai berada pada diagonal utama. Matriks tersebut menunjukkan bahwa dari 10 sampel *setosa*, 10 sampel *versicolor*, dan 10 sampel *virginica* pada data uji, semuanya berhasil diklasifikasikan dengan benar ke kelasnya masing-masing. Hasil ini membuktikan bahwa dataset Iris memiliki kluster antar-spesies yang sangat jelas sehingga algoritma KNN dapat bekerja dengan efektivitas maksimal. Bisa dilihat pada gambar 2 berikut:



Gambar 2 Confusion Matrix

## 2. Implementasi Aplikasi Web

Model yang telah dievaluasi kemudian diimplementasikan dalam sebuah aplikasi web. Aplikasi ini memiliki antarmuka pengguna yang sederhana seperti yang ditunjukkan pada Gambar 3.

**Prediksi Spesies Bunga Iris dengan KNN**

Masukkan ciri-ciri visual bunga iris untuk memprediksi spesiesnya.

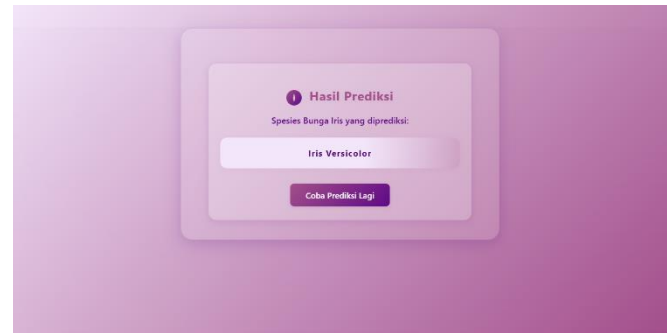
Panjang Kelopak (Sepal Length) dalam cm:   
 Lebar Kelopak (Sepal Width) dalam cm:   
 Panjang Mahkota (Petal Length) dalam cm:   
 Lebar Mahkota (Petal Width) dalam cm:

**Prediksi**

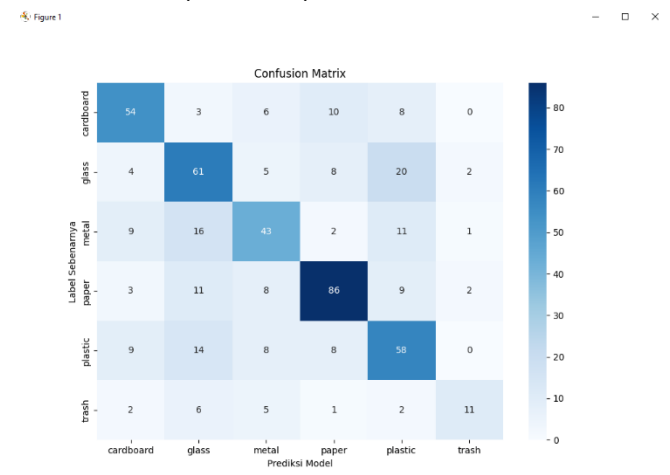
Gambar 3 Tampilan Awal Antarmuka Web

Pengguna dapat memasukkan empat nilai fitur ke dalam formulir yang tersedia. Setelah menekan tombol "Prediksi", data tersebut dikirim ke *backend* Flask. Backend memuat model KNN yang telah disimpan, melakukan prediksi berdasarkan input pengguna, dan

menampilkan hasilnya di halaman baru, seperti yang ditunjukkan pada Gambar 4



Gambar 4 Tampilan Hasil pada Web



Gambar 1 Confusion Matrix Hasil Prediksi Model

Aplikasi ini berhasil membungkus fungsionalitas model *machine learning* ke dalam sebuah produk yang dapat digunakan secara praktis, memenuhi tujuan utama dari penelitian ini.

## **SIMPULAN DAN SARAN**

**1. Simpulan** Berdasarkan hasil penelitian, dapat disimpulkan bahwa algoritma K-Nearest Neighbors (KNN) sangat cocok dan efektif untuk tugas klasifikasi spesies bunga Iris, dengan pencapaian akurasi 100% pada data uji. Implementasi model ini ke dalam sebuah aplikasi web menggunakan *framework* Flask telah berhasil dilakukan, menghasilkan sebuah sistem prediktif yang interaktif, akurat, dan mudah diakses oleh pengguna.

**2. Saran** Untuk pengembangan di masa depan, beberapa hal dapat dieksplorasi lebih lanjut. Pertama, mencoba menerapkan algoritma klasifikasi lain seperti Support Vector Machine (SVM) atau Random Forest untuk membandingkan kinerjanya dengan KNN. Kedua, aplikasi dapat dikembangkan lebih lanjut untuk menerima input berupa gambar bunga, di mana sistem akan melakukan ekstraksi fitur secara otomatis sebelum melakukan klasifikasi. Terakhir, melakukan deployment aplikasi ke platform cloud publik untuk ketersediaan yang lebih luas.

## **PUSTAKA ACUAN**

### **Buku**

Grinberg, M. (2018). Flask Web Development

### **Jurnal**

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python

### **Lain – lain**

-

**LINK YOUTUBE :**

<https://youtu.be/nLGmtbKXZ3o>

**LINK GITHUB :**

<https://github.com/jihadakbar911/tugas9praktikumai>

**LINK WEB YANG  
SUDAH DI-DEPLOY :**

<https://prediksibungairis.up.railway.app/>