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

**Jihad Akbar1**.

***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**

Pertambahan jumlah penduduk dan perubahan gaya hidup masyarakat modern telah menyebabkan peningkatan volume sampah secara signifikan. Pengelolaan sampah yang tidak efektif menjadi salah satu masalah lingkungan yang serius di banyak negara, termasuk Indonesia. Salah satu tahapan krusial dalam pengelolaan sampah adalah proses pemilahan, yang bertujuan untuk memisahkan sampah berdasarkan jenisnya agar dapat didaur ulang atau diolah lebih lanjut dengan tepat. Saat ini, proses pemilahan sebagian besar masih dilakukan secara manual, yang cenderung lambat, tidak efisien, dan memiliki risiko kesehatan bagi para pekerjanya.

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*), khususnya dalam bidang visi komputer (*Computer Vision*), menawarkan solusi inovatif untuk mengatasi masalah ini. Pemanfaatan *machine learning* untuk klasifikasi gambar dapat mengotomatisasi proses pemilahan sampah. Algoritma klasifikasi dapat dilatih untuk mengenali dan membedakan berbagai jenis sampah dari sebuah gambar digital.

Dalam penelitian ini, algoritma yang dipilih adalah **Random Forest**. Algoritma ini merupakan salah satu metode *ensemble learning* yang bekerja dengan membangun banyak pohon keputusan (*decision trees*) dan menggabungkan hasilnya untuk mendapatkan prediksi yang lebih akurat dan stabil. [Random Forest dipilih karena kemampuannya yang baik dalam menangani data berdimensi tinggi seperti data piksel gambar dan lebih tahan terhadap *overfitting* dibandingkan dengan satu pohon keputusan tunggal.](https://www.google.com/search?q=)

Tujuan dari penelitian ini adalah untuk merancang, membangun, dan menguji sebuah sistem aplikasi berbasis web yang fungsional untuk klasifikasi jenis sampah. Sistem ini menggunakan model *machine learning* yang telah dilatih dengan algoritma Random Forest dan diimplementasikan menggunakan *framework* Flask, sehingga dapat diakses oleh pengguna untuk melakukan klasifikasi secara *real-time* dengan mengunggah gambar sampah.

**METODA**

Penelitian ini dilakukan melalui beberapa tahapan yang sistematis, meliputi pengumpulan dataset, preprocessing data, perancangan dan pelatihan model, hingga implementasi ke dalam aplikasi web.

Dataset yang digunakan dalam penelitian ini adalah "Garbage Classification" yang bersumber dari platform Kaggle. Dataset ini terdiri dari total 2.527 gambar yang terbagi ke dalam enam kategori (kelas) yang berbeda, yaitu *cardboard* (kardus), *glass* (kaca), *metal* (logam), *paper* (kertas), *plastic* (plastik), dan *trash* (sampah organik/lainnya).

Sebelum data gambar digunakan untuk melatih model, dilakukan serangkaian tahap **preprocessing**. Setiap gambar dikonversi menjadi format *grayscale* (skala abu-abu) untuk mengurangi kompleksitas komputasi. Selanjutnya, semua gambar diubah ukurannya menjadi dimensi yang seragam, yaitu 128x128 piksel. Gambar yang telah berukuran seragam kemudian diratakan (*flattened*) menjadi sebuah vektor satu dimensi. Tahap terakhir adalah normalisasi, di mana nilai intensitas setiap piksel (yang awalnya berada di rentang 0-255) diubah skalanya menjadi rentang 0-1.

Model klasifikasi dibangun menggunakan algoritma **Random Forest Classifier** dari library **Scikit-Learn** di Python. Dataset yang telah diproses kemudian dibagi menjadi dua bagian: 80% sebagai data latih (*training data*) dan 20% sebagai data uji (*testing data*). Model dilatih menggunakan data latih untuk mempelajari pola dari setiap kelas sampah. Arsitektur sistem akhir berupa aplikasi web yang dikembangkan dengan *framework* Flask. Aplikasi ini menyediakan antarmuka bagi pengguna untuk mengunggah gambar, yang kemudian akan diproses oleh model di sisi *backend* untuk menghasilkan prediksi kelas sampah.

**HASIL DAN PEMBAHASAN**

1. Evaluasi Kinerja Model

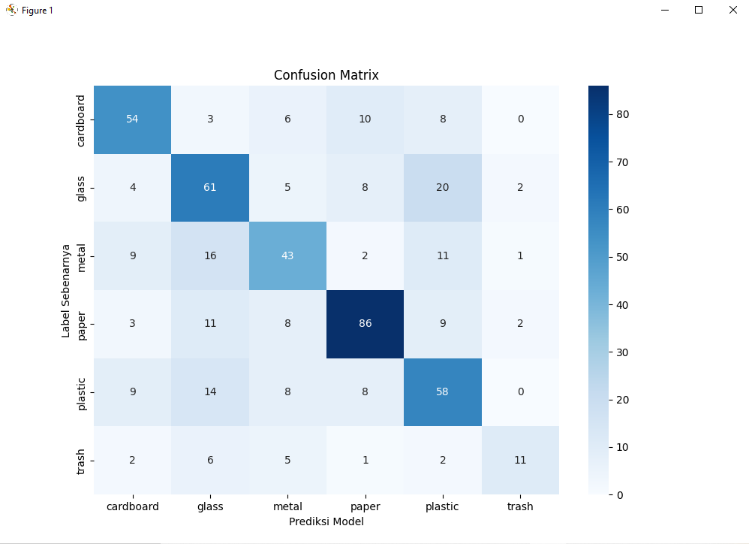
[Setelah proses pelatihan, model dievaluasi menggunakan data uji yang belum pernah dilihat sebelumnya.](https://www.google.com/search?q=) Kinerja model diukur menggunakan beberapa metrik, yaitu akurasi, *precision*, *recall*, dan *F1-score*, yang dirangkum dalam laporan klasifikasi pada Tabel 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kategori | Precision | Recall | F1-Score | Support |
| Cardboard | 0.55 | 0.67 | 0.67 | 81 |
| Glass | 0.67 | 0.61 | 0.58 | 100 |
| Metal | 0.57 | 0.52 | 0.55 | 82 |
| Paper | 0.75 | 0.72 | 0.74 | 119 |
| Plastic | 0.54 | 0.60 | 0.57 | 97 |
| Trash | 0.69 | 0.41 | 0.51 | 27 |
| Accuracy |  |  | 0.62 | 506 |
| Macro avg | 0.63 | 0.59 | 0.60 | 506 |
| Weighted avg | 0.62 | 0.62 | 0.62 | 506 |

Tabel 1.

Dari Tabel 1, terlihat bahwa model mencapai akurasi keseluruhan sebesar **62%**. Hasil ini menunjukkan bahwa model memiliki kemampuan dasar untuk klasifikasi, namun masih terdapat ruang yang signifikan untuk perbaikan. Kelas **'paper'** menunjukkan performa terbaik dengan F1-Score tertinggi (0.74), sementara kelas **'trash'** memiliki performa terendah (0.51). Rendahnya nilai *recall* pada kelas 'trash' (0.41) mengindikasikan bahwa model seringkali gagal mengenali sampah kategori 'trash' dan salah mengklasifikasikannya sebagai kategori lain. Performa yang bervariasi antar kelas ini menguatkan perlunya analisis lebih lanjut terhadap dataset dan model.

Untuk analisis kesalahan yang lebih detail, disajikan *Confusion Matrix* pada Gambar 1.



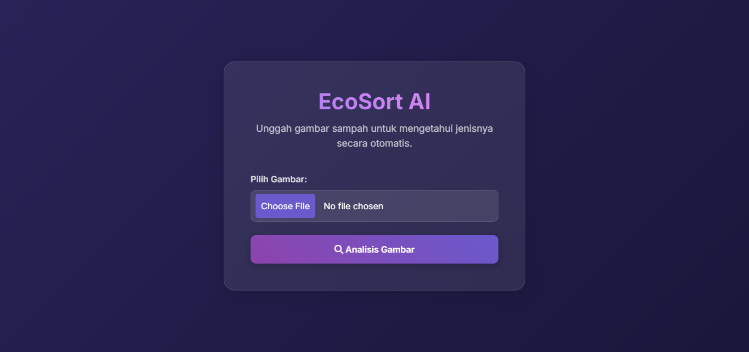
Gambar 1 *Confusion Matrix* Hasil Prediksi Model

1. Analisis Overfitting dan Generalisasi

Meskipun model menunjukkan kinerja yang moderat pada data uji, sebuah pengujian lanjutan dilakukan dengan menggunakan gambar-gambar baru yang diambil secara mandiri di luar dataset. Hasilnya menunjukkan bahwa model seringkali melakukan kesalahan klasifikasi pada gambar-gambar baru tersebut. Fenomena ini mengindikasikan bahwa model mengalami **overfitting**. Model menjadi terlalu "hafal" dengan karakteristik spesifik dari data latih (seperti pencahayaan dan latar belakang yang seragam) dan kurang mampu melakukan generalisasi terhadap data dengan kondisi yang berbeda. Kurangnya keragaman dalam dataset asli diidentifikasi sebagai penyebab utama dari masalah ini.

1. Implementasi Aplikasi Web

Model yang telah dilatih berhasil diimplementasikan ke dalam sebuah aplikasi web menggunakan Flask. Aplikasi ini memiliki antarmuka pengguna yang sederhana dan responsif berkat penggunaan Bootstrap. Pengguna dapat dengan mudah mengakses halaman utama untuk mengunggah gambar sampah (Gambar 2). Setelah gambar diunggah, aplikasi akan memprosesnya dan menampilkan halaman hasil yang berisi gambar yang diunggah beserta hasil prediksi kelasnya (Gambar 3).



 **Gambar 2** Tampilan Halaman Utama Aplikasi Web

**Gambar 3** Tampilan Halaman Hasil Prediksi

**SIMPULAN DAN SARAN**

Berdasarkan hasil penelitian, dapat disimpulkan bahwa algoritma Random Forest berhasil diimplementasikan untuk membangun model klasifikasi jenis sampah dengan **tingkat akurasi sebesar 62% pada data uji**. Model tersebut juga telah berhasil diintegrasikan ke dalam sebuah aplikasi berbasis web menggunakan Flask yang fungsional dan mudah digunakan. Kelemahan utama dari model yang dihasilkan adalah akurasi yang masih moderat dan kecenderungannya untuk **overfitting**, yang menyebabkan penurunan performa ketika dihadapkan pada data baru yang memiliki karakteristik berbeda dari data latih.

Untuk pengembangan selanjutnya, disarankan untuk menerapkan teknik **augmentasi data** (seperti rotasi, zoom, dan flip) pada dataset latih. Teknik ini bertujuan untuk meningkatkan keragaman data secara artifisial. Selain itu, **optimasi hyperparameter** pada model Random Forest, seperti menyesuaikan max\_depth dan n\_estimators, juga perlu dilakukan untuk mencari konfigurasi terbaik. Kombinasi dari kedua teknik ini diharapkan dapat membantu model untuk belajar fitur yang lebih robust, sehingga meningkatkan akurasi dan kemampuan generalisasinya di masa depan.

**PUSTAKA ACUAN**

**Buku**

Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd ed.). O'Reilly Media.

**Jurnal**

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

**Lain – lain**

Kaggle. (2018). *Garbage Classification Dataset.*. <https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification>

**LINK YOUTUBE :** <https://youtu.be/lMkgQGu-_XM>

**LINK GITHUB :**

<https://github.com/jihadakbar911/tugas8praktikumai>

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

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