JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 7, Desember Tahun 2025
ISSN 3025-0919 (media online)
Hal 2083-2087

Penerapan *Decision Tree* dalam *Data Mining* menggunakan Bahasa Pemrograman Python

Agung Adriansyah[1], Data Fitri[2], Fauzi[3], Farhan[4], Juliana[5], Khafi[6]

[1,2,3,4,5,6]Teknik Informatika Universitas Pamulang, Tangerang Selatan, Banten Indonesia
Email : [1]agungadriansyah96@gmail.com, [2]khafiansach@gmail.com, [3]Fauzi9mm@gmail.com, [4]farhhana075@gmail.com, [5]anajulianawulandari25@gmail.com, [6]dosen00848@unpam.ac.id

Abstrak−*Data mining* merupakan proses ekstraksi informasi dan pola dari kumpulan data besar yang dapat digunakan untuk mendukung pengambilan keputusan. Salah satu metode populer dalam data mining adalah *Decision Tree*, yang digunakan untuk klasifikasi dan prediksi. Penelitian ini bertujuan untuk mengimplementasikan algoritma Decision Tree menggunakan bahasa pemrograman Python dengan *library scikit-learn*, serta menganalisis performa model melalui evaluasi akurasi. Hasil penelitian menunjukkan bahwa *Decision Tree* mampu mengklasifikasikan data dengan akurasi yang memuaskan dan dapat menjadi alat bantu efektif dalam pengambilan keputusan berbasis data.

Kata Kunci : Data; *Decision Tree*; Python; *Scikit-learn*; Klasifikasi

Abstract−*Data mining is the process of extracting useful information and patterns from large datasets to support decision-making. One of the most popular methods in data mining is the Decision Tree, which is commonly used for classification and prediction tasks. This study aims to implement the Decision Tree algorithm using the Python programming language and the Scikit-learn library, as well as to analyze the model's performance through accuracy evaluation. The results show that the Decision Tree can classify data with satisfactory accuracy and can serve as an effective tool for data-driven decision-making.*

Keywords: *Data Mining; Decision Tree; Python; Scikit-learn; Classification*

1. PENDAHULUAN

Pada era digital saat ini, jumlah data yang dihasilkan dan dikumpulkan oleh berbagai sektor, seperti bisnis, pendidikan, kesehatan, dan pemerintahan, meningkat secara signifikan. Data tersebut memiliki potensi besar untuk memberikan informasi berharga yang dapat digunakan dalam pengambilan keputusan strategis. Namun, tanpa adanya metode analisis yang tepat, data dalam jumlah besar hanya akan menjadi tumpukan informasi yang tidak berguna. Oleh karena itu, diperlukan suatu proses yang mampu mengekstraksi pengetahuan dan pola tersembunyi dari data tersebut. Proses ini dikenal dengan istilah data mining.

Data mining merupakan bagian dari bidang *Knowledge Discovery in Databases (KDD)* yang berfokus pada penggalian pola, hubungan, dan informasi baru dari data besar menggunakan berbagai teknik statistik, matematika, dan pembelajaran mesin (*machine learning*). Salah satu metode yang paling banyak digunakan dalam data mining adalah *Decision Tree* (pohon keputusan). Metode ini populer karena kemampuannya menghasilkan model yang mudah dipahami, bersifat interpretatif, dan memiliki performa yang baik untuk tugas klasifikasi maupun prediksi.

Decision Tree bekerja dengan cara membagi dataset menjadi beberapa subset berdasarkan atribut tertentu hingga mencapai keputusan akhir. Setiap percabangan dalam pohon mewakili suatu kondisi, sedangkan setiap daun *(leaf)* mewakili hasil klasifikasi. Keunggulan utama dari algoritma ini adalah kemampuannya dalam menangani data kategorikal maupun numerik, serta kemudahan dalam memvisualisasikan hasilnya sehingga dapat membantu pengguna non-teknis memahami logika di balik prediksi model.

Dalam penelitian ini, algoritma *Decision Tree* diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan *library Scikit-learn*, yang menyediakan berbagai fungsi untuk pembangunan model pembelajaran mesin. Implementasi dilakukan menggunakan dataset Iris, yang merupakan salah satu dataset standar untuk pengujian algoritma klasifikasi. Penelitian ini juga bertujuan untuk mengevaluasi performa model melalui pengukuran akurasi, *confusion matrix*, dan visualisasi struktur pohon keputusan.

Diharapkan, hasil penelitian ini dapat menunjukkan bahwa algoritma *Decision Tree* mampu memberikan performa klasifikasi yang baik serta dapat dijadikan alat bantu efektif dalam proses pengambilan keputusan berbasis data *(data-driven decision making)*.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 7, Desember Tahun 2025
ISSN 3025-0919 (media online)
Hal 2083-2087

2. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahap:

1. **Pengumpulan Data**: Dataset diperoleh dari sumber terbuka (misal: dataset iris atau dataset pelanggan).
2. **Preprocessing Data**: Meliputi pembersihan data, penanganan *missing value*, dan normalisasi bila diperlukan.
3. **Implementasi *Decision Tree***: Menggunakan Python dan *library scikit-learn* untuk membangun model klasifikasi.
4. **Evaluasi Model**: Menggunakan metrik akurasi, *confusion matrix*, dan visualisasi pohon keputusan.

2.1 Tujuan

Tujuan dari jurnal berjudul "Penerapan *Decision Tree* dalam *Data Mining* menggunakan Bahasa Pemrograman Python" adalah:

Untuk mengimplementasikan algoritma *Decision Tree* menggunakan bahasa pemrograman Python (dengan *library Scikit-learn*) serta menganalisis performa model melalui evaluasi akurasi.

Secara lebih rinci, tujuan penelitian ini mencakup:

1. Menerapkan algoritma *Decision Tree* sebagai metode klasifikasi dalam data mining.
2. Menguji performa model menggunakan metrik seperti akurasi dan *confusion matrix*.
3. Memvisualisasikan hasil model *Decision Tree* agar mudah dipahami dan diinterpretasikan.
4. Menunjukkan efektivitas *Decision Tree* dalam mendukung proses pengambilan keputusan berbasis data *(data-driven decision making)*.

Singkatnya, jurnal ini bertujuan untuk membuktikan bahwa algoritma *Decision Tree* dapat menjadi alat yang efektif, akurat, dan mudah digunakan untuk analisis dan klasifikasi data menggunakan Python.

3. ANALISA DAN PEMBAHASAN

Berikut contoh implementasi *Decision Tree* menggunakan dataset Iris:

```python
# Import library
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn import tree
import matplotlib.pyplot as plt

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target
```

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 7, Desember Tahun 2025
ISSN 3025-0919 (media online)
Hal 2083-2087

```
# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Buat model Decision Tree
model = DecisionTreeClassifier(criterion='entropy', random_state=42)
model.fit(X_train, y_train)

# Prediksi
y_pred = model.predict(X_test)

# Evaluasi
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
print("Akurasi:", accuracy)
print("Confusion Matrix:\n", cm)
```

```
# Visualisasi Decision Tree
plt.figure(figsize=(12,8))
tree.plot_tree(model, feature_names=iris.feature_names, class_names=iris.target_names, filled=True)
plt.show()
```

Copy code

3.1 Tahap 1: *Import Library*

Pada bagian awal, program mengimpor beberapa *library* penting, yaitu:

a. pandas untuk mengelola data,
b. load_iris untuk memanggil dataset iris,
c. train_test_split untuk membagi data menjadi data latih dan data uji,
d. DecisionTreeClassifier untuk membuat model pohon keputusan,
e. accuracy_score dan confusion_matrix untuk mengevaluasi performa model,
f. tree dan matplotlib.pyplot untuk menampilkan visualisasi hasil pohon keputusan.
g. Import ini adalah langkah awal yang penting agar semua fungsi yang diperlukan tersedia dalam program.

3.2 Tahap 2: Memuat Dataset

```
iris = load_iris()
X = iris.data
y = iris.target
```

Dataset Iris dimuat langsung dari *Scikit-learn*. Dataset ini berisi 150 data bunga iris dengan empat fitur:

a. panjang sepal,
b. lebar sepal,
c. panjang petal,
d. lebar petal.

Sedangkan y berisi label kelas dari tiga jenis bunga iris: setosa, versicolor, dan virginica. Variabel X menampung data fitur, sedangkan y menampung target klasifikasi.

3.3 Tahap 3: Membagi Data

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Data dibagi menjadi dua bagian:

a. 70% untuk pelatihan (training),
b. 30% untuk pengujian (testing).

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 7, Desember Tahun 2025
ISSN 3025-0919 (media online)
Hal 2083-2087

Parameter random_state=42 digunakan agar pembagian data selalu konsisten setiap kali program dijalankan. Langkah ini penting agar model dapat dilatih dan diuji secara terpisah, sehingga hasil akurasi lebih objektif.

3.4 Tahap 4: Membuat dan Melatih Model *Decision Tree*

```
model = DecisionTreeClassifier(criterion='entropy', random_state=42)
model.fit(X_train, y_train)
```

Model pohon keputusan dibuat menggunakan algoritma DecisionTreeClassifier. Kriteria yang digunakan adalah 'entropy', yang berarti pemilihan percabangan dilakukan berdasarkan perhitungan Information Gain. Kemudian, model dilatih menggunakan data latih (X_train, y_train) agar dapat mengenali pola antara fitur dan kelas target.

3.5 Tahap 5: Melakukan Prediksi

```
y_pred = model.predict(X_test)
```

Setelah model dilatih, langkah berikutnya adalah menguji kemampuannya dalam memprediksi kelas pada data uji (X_test). Hasil prediksi disimpan dalam variabel y_pred dan nantinya dibandingkan dengan y_test (data sebenarnya).

3.6 Tahap 6: Evaluasi Model

```
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
print("Akurasi:", accuracy)
print("Confusion Matrix:\n", cm)
```

Hasil prediksi kemudian dievaluasi dengan dua metrik utama:

1. Akurasi (*accuracy*) → menunjukkan persentase prediksi yang benar dibandingkan total data uji.
2. *Confusion Matrix* → menampilkan distribusi hasil prediksi antara kelas yang benar dan salah.

Biasanya, pada dataset Iris, model *Decision Tree* dapat mencapai akurasi di atas 90%, menandakan performa yang baik untuk klasifikasi sederhana.

3.7 Tahap 7: Visualisasi Pohon Keputusan

```
plt.figure(figsize=(12,8))
tree.plot_tree(model, feature_names=iris.feature_names, class_names=iris.target_names, filled=True)
plt.show()
```

Langkah terakhir adalah menampilkan visualisasi struktur pohon keputusan. Setiap node pada pohon menunjukkan kondisi atau aturan pemisahan berdasarkan nilai fitur tertentu. Parameter filled=True memberikan warna berbeda untuk tiap kelas agar hasil lebih mudah dibaca. Visualisasi ini sangat membantu untuk memahami bagaimana model mengambil keputusan berdasarkan fitur-fitur data.

Berdasarkan hasil implementasi, model *Decision Tree* menunjukkan akurasi yang tinggi (biasanya >90% pada dataset Iris). Confusion matrix menunjukkan distribusi prediksi yang sesuai dengan kelas sebenarnya, sehingga model ini efektif untuk klasifikasi data sederhana. Visualisasi pohon keputusan mempermudah interpretasi aturan yang digunakan model untuk mengambil keputusan.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 7, Desember Tahun 2025
ISSN 3025-0919 (media online)
Hal 2083-2087

4. KESIMPULAN

Decision Tree merupakan algoritma yang efektif dan mudah diinterpretasikan dalam data mining untuk tugas klasifikasi. Implementasi dengan Python dan scikit-learn memberikan fleksibilitas tinggi dan hasil yang akurat. Model ini dapat diterapkan pada berbagai jenis dataset untuk mendukung pengambilan keputusan berbasis data.

REFERENCES

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

Larose, D. T., & Larose, C. D. (2014). Discovering Knowledge in Data: An Introduction to Data Mining. Wiley.

Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Prasetyo, E. (2012). Data Mining: Konsep dan Aplikasi Menggunakan MATLAB. Yogyakarta: Andi Offset.

Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81–106.

Sudarmaji, & Astuti, W. (2021). Penerapan Algoritma Decision Tree untuk Klasifikasi Data Menggunakan Python. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 8(2), 145–153.

Witten, I. H., Frank, E., & Hall, M. A. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.