

**Nama : Putri Amalia**

**NIM : 231011401598**

**Kelas : 05TPLE013**

## **LAPORAN UAS KECERDASAN BUATAN**

### **IMPLEMENTASI DECISION TREE**

Studi Kasus: Dataset Iris

## **BAB I**

### **PENDAHULUAN**

Perkembangan teknologi informasi saat ini mendorong pemanfaatan data dalam jumlah besar untuk membantu proses pengambilan keputusan. Salah satu bidang yang berkembang pesat dalam pemanfaatan data tersebut adalah **Machine Learning**, yaitu cabang dari kecerdasan buatan yang memungkinkan sistem komputer belajar dari data tanpa diprogram secara eksplisit.

Salah satu algoritma Machine Learning yang banyak digunakan adalah **Decision Tree**. Algoritma ini sering dipilih karena memiliki konsep yang sederhana, mudah dipahami, serta dapat divisualisasikan dalam bentuk pohon keputusan. Decision Tree dapat digunakan untuk berbagai permasalahan, baik klasifikasi maupun regresi.

Pada tugas Ujian Akhir Semester (UAS) ini, dilakukan implementasi algoritma Decision Tree menggunakan **Dataset Iris**. Dataset Iris dipilih karena merupakan dataset klasik yang sering digunakan dalam pembelajaran machine learning, memiliki struktur data yang jelas, serta cocok untuk permasalahan klasifikasi multikelas.

Tujuan dari penyusunan laporan ini adalah untuk memahami konsep dasar Decision Tree, mengimplementasikan algoritma Decision Tree menggunakan bahasa pemrograman Python dengan library scikit-learn, serta menganalisis performa model berdasarkan hasil evaluasi yang diperoleh.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Decision Tree**

Decision Tree adalah algoritma supervised learning yang digunakan untuk menyelesaikan permasalahan klasifikasi dan regresi. Algoritma ini bekerja dengan cara membagi dataset ke dalam beberapa bagian berdasarkan kondisi tertentu hingga mencapai keputusan akhir. Proses pengambilan keputusan pada Decision Tree digambarkan dalam bentuk struktur pohon yang terdiri dari node dan cabang.

Setiap keputusan yang diambil didasarkan pada nilai fitur tertentu yang dipilih berdasarkan kriteria tertentu, seperti Gini Index atau Entropy. Karena strukturnya yang menyerupai pohon, Decision Tree mudah dipahami oleh manusia dan sering digunakan dalam sistem pendukung keputusan.

#### **2.2 Konsep Dasar pada Decision Tree**

##### **1. Node**

Node merupakan titik pada pohon keputusan yang merepresentasikan suatu fitur atau kondisi tertentu. Node digunakan sebagai tempat terjadinya proses pengambilan keputusan.

##### **2. Root**

Root adalah node paling atas pada pohon keputusan. Node ini menjadi titik awal proses klasifikasi dan biasanya merupakan fitur paling penting yang digunakan untuk membagi data.

##### **3. Leaf**

Leaf adalah node terakhir pada pohon keputusan. Node ini tidak memiliki cabang lagi dan berisi hasil akhir berupa kelas atau nilai prediksi.

##### **4. Splitting**

Splitting adalah proses pemisahan data ke dalam beberapa cabang berdasarkan nilai dari suatu fitur. Tujuan splitting adalah untuk meningkatkan homogenitas data dalam setiap cabang sehingga hasil prediksi menjadi lebih akurat.

## 5. Pruning

Pruning adalah proses pemangkasan cabang pada pohon keputusan. Teknik ini digunakan untuk mengurangi kompleksitas model dan mencegah terjadinya overfitting, yaitu kondisi di mana model terlalu menyesuaikan diri dengan data training.

### 2.3 Perbedaan Decision Tree, Random Forest, dan Gradient Boosting

- Decision Tree menggunakan satu pohon keputusan untuk melakukan prediksi. Metode ini sederhana dan mudah diinterpretasikan, namun rentan terhadap overfitting.
- Random Forest merupakan metode ensemble yang menggabungkan banyak Decision Tree. Setiap pohon dibangun dari subset data dan fitur yang berbeda, kemudian hasil prediksi digabungkan untuk menghasilkan keputusan akhir yang lebih stabil.
- Gradient Boosting adalah metode ensemble yang membangun pohon keputusan secara bertahap. Setiap model baru dibuat untuk memperbaiki kesalahan dari model sebelumnya sehingga menghasilkan performa yang lebih baik.

### 2.4 Kelebihan dan Kekurangan Tree-Based Methods

- Mudah dipahami dan diinterpretasikan
- Dapat menangani data numerik dan kategorik
- Tidak memerlukan normalisasi data
- Dapat digunakan untuk klasifikasi dan regresi
- Rentan terhadap overfitting jika pohon terlalu dalam
- Sensitif terhadap perubahan kecil pada data
- Decision Tree tunggal kurang optimal untuk dataset besar dan kompleks

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Dataset**

Dataset yang digunakan dalam penelitian ini adalah **Dataset Iris**. Dataset ini terdiri dari 150 data dengan empat fitur utama, yaitu sepal length, sepal width, petal length, dan petal width. Dataset Iris memiliki tiga kelas target, yaitu Iris setosa, Iris versicolor, dan Iris virginica.

Dataset ini digunakan untuk permasalahan klasifikasi multikelas dan sangat sesuai untuk menguji performa algoritma Decision Tree.

#### **3.2 Preprocessing Data**

Tahap preprocessing bertujuan untuk menyiapkan data sebelum digunakan dalam pemodelan. Pada Dataset Iris tidak ditemukan missing value dan seluruh fitur bersifat numerik, sehingga tidak diperlukan proses penanganan missing value maupun encoding data. Data dapat langsung digunakan pada tahap pemodelan.

#### **3.3 Pembagian Data**

Dataset dibagi menjadi dua bagian, yaitu data training dan data testing. Data training digunakan untuk melatih model, sedangkan data testing digunakan untuk menguji performa model. Pada penelitian ini, data dibagi dengan proporsi 80% untuk data training dan 20% untuk data testing.

#### **3.4 Pembuatan Model Decision Tree**

Model yang digunakan adalah **Decision Tree Classifier** dengan parameter utama `criterion = gini` dan `max_depth = 3`. Pemilihan parameter ini bertujuan untuk mengontrol kompleksitas model agar tidak terjadi overfitting serta tetap menghasilkan performa yang optimal.

## BAB IV

### HASIL DAN ANALISIS

#### 4.1 Hasil Eksperimen

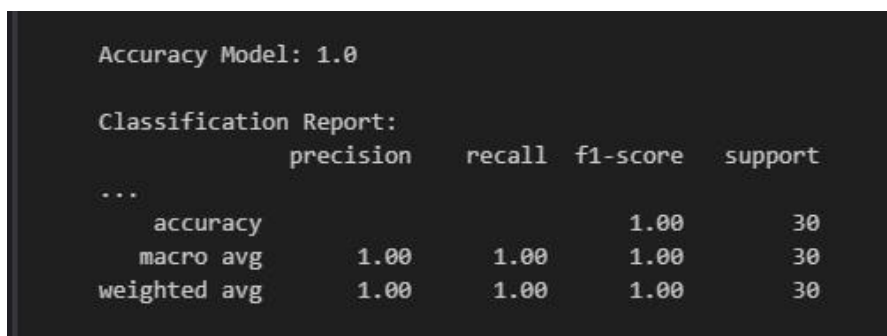
Pada penelitian ini, pengujian model dilakukan menggunakan algoritma **Decision Tree** dengan dataset **Iris**. Data dibagi menjadi data training dan data testing dengan perbandingan 80% untuk data training dan 20% untuk data testing, sehingga diperoleh 120 data training dan 30 data testing.

Berdasarkan hasil pengujian yang telah dilakukan, model Decision Tree menghasilkan nilai **accuracy sebesar 1.0 atau 100%**. Hal ini menunjukkan bahwa seluruh data pada data testing berhasil diklasifikasikan dengan benar oleh model.

Selain itu, hasil **classification report** menunjukkan bahwa nilai **precision**, **recall**, dan **f1-score** pada seluruh kelas memiliki nilai **1.00**. Nilai ini menandakan bahwa model memiliki performa yang sangat baik dalam membedakan setiap kelas pada dataset Iris. Tidak ditemukan kesalahan klasifikasi pada data uji, sehingga model dapat dikatakan bekerja secara optimal pada dataset yang digunakan.

Keberhasilan model dalam mencapai akurasi yang tinggi dipengaruhi oleh karakteristik dataset Iris yang memiliki pola data yang jelas dan fitur yang saling terpisah dengan baik, sehingga algoritma Decision Tree mampu mempelajari aturan klasifikasi dengan efektif.

**Gambar 4.1 Hasil Evaluasi Model Decision Tree**



```
Accuracy Model: 1.0

Classification Report:
              precision    recall  f1-score   support

...
  accuracy                1.00      1.00      1.00        30
  macro avg              1.00      1.00      1.00        30
  weighted avg           1.00      1.00      1.00        30
```

Accuracy Model: 1.0				
Classification Report:				
	precision	recall	f1-score	support
...				
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

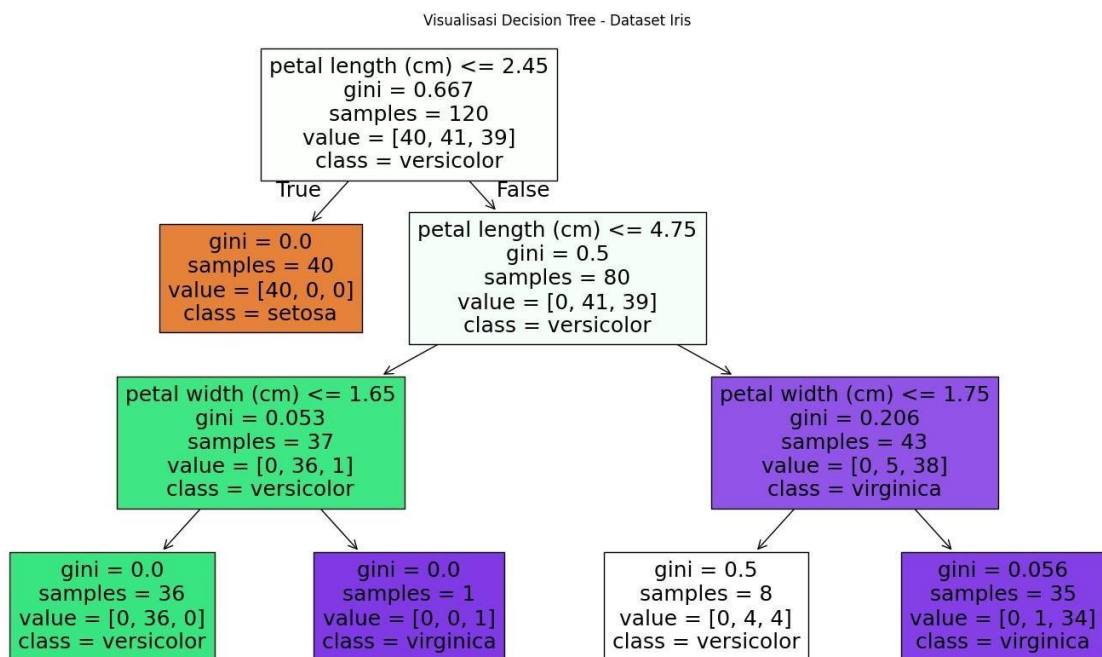
## 4.2 Visualisasi Pohon Keputusan

Visualisasi pohon keputusan dilakukan untuk melihat proses pengambilan keputusan yang dilakukan oleh model Decision Tree. Pada visualisasi tersebut, setiap node menunjukkan fitur yang digunakan sebagai dasar pemisahan data, seperti panjang dan lebar sepal maupun petal.

Pohon keputusan memperlihatkan bagaimana data dibagi secara bertahap hingga mencapai node daun (leaf) yang merepresentasikan kelas akhir dari bunga Iris. Visualisasi ini menunjukkan kelebihan algoritma Decision Tree, yaitu kemampuannya dalam memberikan model yang mudah dipahami dan diinterpretasikan oleh pengguna.

Dengan adanya visualisasi pohon keputusan, proses klasifikasi menjadi lebih transparan karena pengguna dapat mengetahui alasan model dalam menentukan kelas suatu data berdasarkan nilai fitur yang dimiliki.

**Gambar 4.2 Visualisasi Pohon Keputusan**



## **BAB V**

### **KESIMPULAN**

Berdasarkan hasil implementasi dan analisis yang telah dilakukan, dapat disimpulkan bahwa algoritma Decision Tree sangat efektif digunakan untuk melakukan klasifikasi pada Dataset Iris. Model yang dibangun menghasilkan tingkat akurasi yang tinggi serta nilai precision, recall, dan f1-score yang optimal. Selain memiliki performa yang baik, Decision Tree juga mudah diinterpretasikan melalui visualisasi pohon keputusan. Oleh karena itu, tree-based methods cocok digunakan untuk permasalahan klasifikasi dengan dataset yang terstruktur dan berukuran kecil hingga menengah.