

**ANALISIS EVALUASI KELAYAKAN MOBIL DENGAN METODE
DECISION TREE**



Dosen Mata Kuliah : Dr. Ir. Ricky Eka Putra, S.Kom, M.Kom.

Disusun Oleh :

Kelompok 3

Rayyan (23051204385)

Nicholas Rahadiyan Wijaya (23051204390)

**UNIVERSITAS NEGERI SURABAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK INFORMATIKA
PRODI TEKNIK INFORMATIKA
2024**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga terselesaikannya penelitian klasifikasi data evaluasi mobil menggunakan algoritma Decision Tree. Penelitian ini merupakan upaya untuk mengaplikasikan teknologi kecerdasan buatan dalam menganalisis dan memprediksi kelayakan mobil berdasarkan berbagai parameter yang ada. Laporan ini disusun sebagai bagian dari upaya untuk memenuhi tugas akhir dan memahami lebih dalam penerapan metode analisis data modern dalam bidang industri otomotif, khususnya dalam memprediksi evaluasi kelayakan mobil dengan beberapa parameter yang ada.

Di tengah pesatnya perkembangan teknologi dan kebutuhan akan sistem pengambilan keputusan yang cepat dan akurat, metode Decision Tree menawarkan pendekatan yang sederhana namun efektif dalam menganalisis data. Melalui laporan ini, kami berusaha menjelaskan secara komprehensif proses, analisis, dan hasil analisis evaluasi kelayakan mobil dengan menggunakan metode tersebut.

Kami juga berharap laporan ini dapat memberikan wawasan baru bagi pembaca tentang penerapan algoritma Decision Tree dalam menyelesaikan permasalahan nyata di dunia industri maupun teknologi. Kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan untuk menyempurnakan karya di masa mendatang. Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, sehingga laporan ini dapat diselesaikan. Semoga dengan laporan ini bermanfaat bagi pembaca dan dapat menjadi referensi yang berguna di masa depan.

LATAR BELAKANG

Di era digital kontemporer, perkembangan teknologi informasi mengalami percepatan yang signifikan, terutama dalam bidang kecerdasan buatan (Artificial Intelligence). Kemajuan ini telah merambah hampir seluruh sektor, termasuk analisis data dan pengambilan keputusan yang strategis.

Dalam konteks evaluasi mobil, terdapat kompleksitas yang tinggi dalam menentukan kelayakan sebuah kendaraan. Berbagai faktor seperti harga pembelian, biaya perawatan, jumlah pintu, kapasitas penumpang, ukuran bagasi, dan tingkat keamanan menjadi pertimbangan penting. Algoritma Decision Tree hadir sebagai solusi cerdas untuk mengurai kompleksitas ini.

Klasifikasi adalah suatu proses analisis data yang menghasilkan model untuk mewakili kelas-kelas dalam data tersebut (Dzikrina, 2021). Model ini, dikenal sebagai classifier, membentuk kelas-kelas dalam data, seperti yang terjadi pada Decision Tree yang menggambarkan struktur pohon. Teknik klasifikasi, sebagai metode data mining, terbukti bermanfaat dalam mengevaluasi mobil (Maulida, 2018). Penelitian data mining dalam evaluasi mobil sering kali fokus pada penerapan teknik klasifikasi, seperti penggunaan algoritma Decision Tree ID3 untuk memprediksi kelayakan mobil (Hana, 2020).

Python, melalui pustaka seperti scikit-learn, menyediakan alat untuk implementasi algoritma ini dengan mudah dan efisien. Penggunaan Decision Tree dalam evaluasi mobil membantu konsumen membuat keputusan pembelian yang lebih baik dan mendukung produsen dalam meningkatkan kualitas produk mereka. Algoritma ini memberikan solusi efektif dan praktis dalam klasifikasi data, khususnya dalam pengambilan keputusan berbasis data.

TUJUAN PENELITIAN

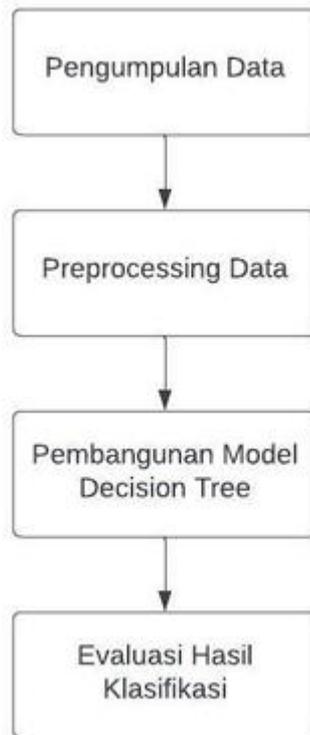
Tujuan penelitian ini adalah untuk mengimplementasikan algoritma Decision Tree dalam klasifikasi data evaluasi mobil dengan memanfaatkan Python sebagai alat pengolahan data dan analisis. Penelitian ini bertujuan untuk mengembangkan model klasifikasi yang mampu secara akurat mengelompokkan mobil berdasarkan atribut tertentu, seperti harga, kapasitas penumpang, keamanan, dan fitur lainnya, menggunakan dataset "Car Evaluation" dari UC Irvine Machine Learning Repository.

Selain itu, penelitian ini bertujuan untuk mengevaluasi performa model dengan mengukur tingkat akurasi, presisi, recall, dan F1-score yang dihasilkan. Dengan menganalisis hasil model pada data pelatihan dan data pengujian, penelitian ini juga berupaya mendeteksi potensi masalah seperti overfitting atau underfitting, serta membandingkan kinerja Decision Tree berdasarkan kriteria Gini Index dan Entropy.

Lebih jauh, penelitian ini bertujuan untuk memberikan solusi berbasis kecerdasan buatan yang efektif dalam mendukung pengambilan keputusan terkait evaluasi mobil, baik untuk membantu konsumen dalam memilih kendaraan yang sesuai dengan kebutuhan mereka, maupun untuk produsen dalam meningkatkan kualitas produk mereka. Dengan menyajikan visualisasi pohon keputusan, penelitian ini diharapkan dapat memberikan wawasan yang lebih mendalam mengenai pola-pola klasifikasi yang ditemukan dalam data.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan deskriptif kuantitatif untuk menganalisis data dan mengembangkan model klasifikasi. Metode deskriptif kuantitatif digunakan untuk menggambarkan fenomena berdasarkan data yang telah terkumpul dan diolah menggunakan angka-angka yang relevan. Berikut adalah tahapan penelitian yang dilakukan:



1. Pengumpulan Data

Dataset yang digunakan adalah *Car Evaluation Dataset* dari UC Irvine Machine Learning Repository. Dataset ini mencakup atribut seperti harga mobil (Buying), biaya perawatan (Maint), jumlah pintu (Doors), kapasitas penumpang (Person), ukuran bagasi (Lug_boot), tingkat keamanan (Safety), dan kelas evaluasi (Class) yang terdiri dari *unacceptable*, *acceptable*, *good*, dan *very good*.

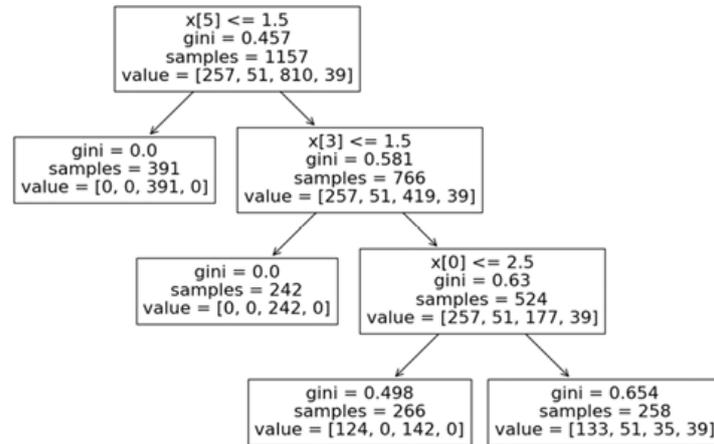
2. Preprocessing Data

- **Import Dataset:** Data diimpor dari sumber eksternal untuk dimasukkan ke dalam program Python.
- **Manipulasi Data:** Langkah ini meliputi penghapusan baris dengan nilai tertentu, penanganan *missing value*, deklarasi atribut, dan pembagian data menjadi *training set* dan *test set*.
- **Visualisasi Data:** Pola-pola data divisualisasikan untuk mempermudah analisis dan pemahaman.

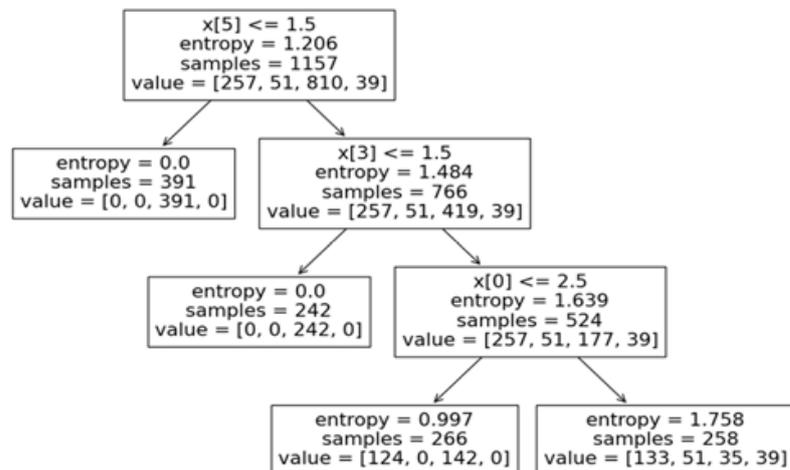
3. Pembangunan Model Decision Tree

Dua model klasifikasi dibangun menggunakan kriteria:

- **Gini Index:** Menggunakan indeks Gini sebagai kriteria untuk pemisahan node pada pohon keputusan.



- **Entropy:** Menggunakan Entropy untuk menghitung Information Gain sebagai dasar pemisahan node.



Model dibuat dengan pustaka Python seperti *scikit-learn*, dengan langkah-langkah:

- Membuat instance model.
- Melatih model dengan *training set*.
- Melakukan prediksi pada *test set*.

4. Evaluasi Model

Evaluasi dilakukan menggunakan metrik seperti akurasi, precision, recall, F1-score, dan Confusion Matrix untuk mengukur performa model.

- *Overfitting* dan *Underfitting*: Model diuji untuk mendeteksi masalah ini dengan membandingkan hasil pada data pelatihan dan pengujian.
- *Accuracy Testing*: Menghitung akurasi data pelatihan dan pengujian untuk kedua kriteria (Gini dan Entropy).

5. Visualisasi Model

Struktur Decision Tree divisualisasikan untuk menunjukkan alur pengambilan keputusan dari akar hingga daun. Visualisasi ini mempermudah interpretasi hubungan antar atribut dan prediksi model.

Metode ini diharapkan memberikan model klasifikasi yang dapat digunakan secara praktis untuk mendukung pengambilan keputusan dalam evaluasi mobil.

PROGRAM

```
9 # URL dataset Car Evaluation
10 url = "https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data"
11
12 # Menentukan kolom dari dataset
13 columns = ['Buying', 'Maint', 'Doors', 'Persons', 'Lug_boot', 'Safety', 'Class']
14
15 # Load dataset langsung dari URL
16 data = pd.read_csv(url, header=None, names=columns)
17
18 # Preprocessing data (Ordinal Encoding)
19 encoder = OrdinalEncoder()
20 for col in data.columns:
21     data[col] = encoder.fit_transform(data[col].values.reshape(-1, 1))
```

- **URL dataset:** Dataset yang digunakan adalah dataset *Car Evaluation* yang diambil dari UCI Machine Learning Repository.
- **Definisi kolom:** Kolom-kolom dari dataset didefinisikan dalam list `columns`.
- **Load data:** Dataset dimuat langsung dari URL dan diberi nama kolom sesuai dengan list yang sudah ditentukan.
- **Ordinal Encoding:** Data kategorikal yang berisi nilai-nilai tekstual seperti "low", "high", "med" diubah menjadi nilai numerik dengan `OrdinalEncoder`. Proses ini penting karena model machine learning hanya bisa bekerja dengan data numerik.
- **Pisahkan fitur dan target:** Data dipisahkan menjadi fitur (X) dan target (y), di mana target adalah kolom `Class`.

```
27 # Membagi dataset menjadi data latih (70%) dan data uji (30%) dengan stratifikasi
28 X_train, X_test, y_train, y_test = train_test_split(
29     X, y, test_size=0.3, random_state=42, stratify=y
30 )
```

- **train_test_split:** Fungsi ini digunakan untuk membagi dataset menjadi dua bagian: data latih (70%) dan data uji (30%).

- **stratify=y**: Stratifikasi memastikan bahwa distribusi target (y) antara data latih dan uji tetap seimbang (proporsional), sehingga tidak terjadi ketidakseimbangan kelas antara kedua set tersebut.

```
31
32 # Menangani class imbalance dengan oversampling menggunakan SMOTE
33 smote = SMOTE(random_state=42)
34 X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
35
```

- **SMOTE (Synthetic Minority Over-sampling Technique)**: Digunakan untuk menangani ketidakseimbangan kelas dalam dataset. SMOTE menciptakan sampel sintetis baru untuk kelas minoritas (kelas yang jarang) dengan cara membuat kombinasi dari sampel yang ada.
- **fit_resample**: Fungsi ini digunakan untuk melakukan resampling pada data latih (X_train dan y_train) dengan SMOTE, menghasilkan dataset yang lebih seimbang antara kelas mayoritas dan minoritas.

```

52 # Evaluasi model
53 print("Classification Report:")
54 print(classification_report(y_test, y_pred, target_names=['unacc', 'acc', 'good', 'vgood'], zero_division=0))
55
56 # Confusion Matrix
57 conf_matrix = confusion_matrix(y_test, y_pred)
58 print("\nConfusion Matrix:\n", conf_matrix)

```

- **DecisionTreeClassifier**: Merupakan model pohon keputusan yang digunakan untuk klasifikasi. Pada model ini, beberapa parameter disesuaikan:
 - `criterion='gini'`: Digunakan untuk memilih fitur yang akan membagi data berdasarkan Gini impurity (nilai yang lebih rendah menunjukkan pembagian yang lebih baik).
 - `max_depth=5`: Membatasi kedalaman pohon untuk menghindari overfitting (pohon yang terlalu dalam akan terlalu kompleks dan cocok dengan noise dalam data).
 - `min_samples_split=5`: Menentukan jumlah minimal sampel yang diperlukan untuk melakukan pemisahan node. Ini mencegah pembagian yang tidak bermakna pada node yang sangat kecil.
 - `min_samples_leaf=5`: Mengharuskan setiap daun memiliki setidaknya 5 sampel, yang juga mencegah pembagian yang terlalu kecil.
 - `class_weight='balanced'`: Memberikan bobot lebih pada kelas minoritas untuk menangani masalah ketidakseimbangan kelas, memastikan model tidak terlalu bias terhadap kelas mayoritas.
 - `random_state=42`: Memastikan bahwa hasil eksperimen konsisten dan dapat direproduksi.

fit(): Digunakan untuk melatih model menggunakan data latih yang sudah di-oversample oleh SMOTE.

```

35
36 # Membuat model Decision Tree dengan pengaturan khusus
37 model = DecisionTreeClassifier(
38     criterion='gini',          # Gunakan kriteria Gini (default)
39     max_depth=5,              # Mengatur max_depth untuk mencegah overfitting
40     min_samples_split=5,      # Menentukan jumlah minimal sampel untuk pembagian
41     min_samples_leaf=5,      # Menghindari overfitting dengan menentukan leaf minimal
42     random_state=42,         # Pastikan hasil konsisten
43     class_weight='balanced'  # Menangani ketidakseimbangan kelas
44 )
45
46 # Melatih model
47 model.fit(X_train_smote, y_train_smote)
48

```

- **model.predict()**: Setelah model dilatih, fungsi `predict` digunakan untuk memprediksi kelas dari data uji (`X_test`).
- **classification_report()**: Fungsi ini memberikan metrik evaluasi seperti presisi, recall, dan F1-score untuk setiap kelas. Ini memberikan gambaran menyeluruh tentang seberapa baik model dalam memprediksi setiap kelas.
 - `target_names=['unacc', 'acc', 'good', 'vgood']`: Menyediakan label untuk setiap kelas.
 - `zero_division=0`: Menghindari pembagian dengan nol saat menghitung metrik evaluasi (misalnya ketika ada kelas yang tidak terdeteksi oleh model).
- **confusion_matrix()**: Fungsi ini memberikan **confusion matrix**, yang menunjukkan jumlah prediksi yang benar dan salah antara kelas-kelas yang ada. Ini membantu untuk mengevaluasi secara lebih rinci kesalahan model dalam memprediksi kelas tertentu.

Kesimpulan

Penelitian ini berhasil mengimplementasikan algoritma Decision Tree untuk mengklasifikasikan kelayakan mobil menggunakan dataset UC Irvine, dengan metode yang mencakup preprocessing data melalui Ordinal Encoding dan SMOTE, pembangunan model Decision Tree dengan kriteria Gini Index dan Entropy, serta evaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score. Dengan mengatur parameter untuk menghindari overfitting dan memberikan bobot seimbang antar kelas, penelitian ini mendemonstrasikan kemampuan algoritma Decision Tree dalam menganalisis kompleksitas data evaluasi mobil secara efektif, memberikan solusi yang berguna baik bagi konsumen maupun produsen dalam pengambilan keputusan.