

## Implementasi Algoritma *Decision Tree* untuk Klasifikasi Data Evaluation Car Menggunakan Python

**Ekin Adhi Guna**

Universitas Negeri Medan

Email: [ekinadhiguna@gmail.com](mailto:ekinadhiguna@gmail.com)

**M. Davin Diza Ghifary**

Universitas Negeri Medan

Email: [Dizadavid03@gmail.com](mailto:Dizadavid03@gmail.com)

**Esra Fransiska Sihombing**

Universitas Negeri Medan

Email: [esra0206002@gmail.com](mailto:esra0206002@gmail.com)

**Age Pius Datubara**

Universitas Negeri Medan

Email: [agepius193@gmail.com](mailto:agepius193@gmail.com)

Jl. William Iskandar Ps. V, Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara 20221

Korespondensi penulis: [ekinadhiguna@gmail.com](mailto:ekinadhiguna@gmail.com)

**Abstract.** *In the digital era, advances in information technology are experiencing rapid growth. One significant development is in the field of artificial intelligence, which has been widely applied in various sectors, including data analysis and decision-making. Researchers in the field of data mining have created various classification algorithms that improve the classification process by utilizing numerical and nominal attributes. Classification is a data analysis process that produces a model to represent the classes in the data, as is the case with Decision Tree which is used to analyze classification and prediction patterns of data and describe the relationship between attribute variable  $x$  and target variable  $y$  in the form of a tree structure. Python, as a popular programming language in artificial intelligence development, provides libraries and frameworks that support the implementation of the Decision Tree algorithm. By applying the Decision Tree algorithm for car evaluation data classification using Python, we can utilize the power of AI to provide effective and efficient solutions in decision-making related to car evaluation. Using the Evaluation Car dataset from UC Irvine Machine Learning Repository, the results showed an accuracy of 81%. Confusion Matrix and classification reports show the model's good performance in making predictions.*

**Keywords:** *Classification, Decision Tree, Python, Google Collaboration*

**Abstrak.** Di era digital saat ini, kemajuan teknologi informasi mengalami pertumbuhan yang pesat. Salah satu perkembangan yang signifikan terjadi dalam bidang kecerdasan buatan (*Artificial Intelligence*), yang telah diterapkan luas di berbagai sektor, termasuk analisis data dan pengambilan keputusan. Para peneliti di bidang data *mining* telah menciptakan beragam algoritma klasifikasi yang meningkatkan proses klasifikasi dengan memanfaatkan atribut numerik dan nominal. Klasifikasi adalah suatu proses analisis data yang menghasilkan model untuk mewakili kelas-kelas dalam data tersebut, seperti yang terjadi pada *Decision Tree* yang digunakan untuk menganalisis klasifikasi dan pola prediksi data serta menggambarkan hubungan antara variabel atribut  $x$  dan variabel target  $y$  dalam bentuk struktur pohon. Python, sebagai bahasa pemrograman populer dalam pengembangan kecerdasan buatan, menyediakan pustaka dan *framework* yang mendukung implementasi algoritma *Decision Tree*. Dengan menerapkan algoritma *Decision Tree* untuk klasifikasi data evaluasi mobil menggunakan Python, kita dapat memanfaatkan kekuatan AI untuk memberikan solusi efektif dan efisien dalam pengambilan keputusan terkait evaluasi mobil. Menggunakan dataset *Evaluation Car* dari UC Irvine Machine Learning Repository, hasil penelitian menunjukkan akurasi sebesar 81%. *Confusion Matrix* dan laporan klasifikasi menunjukkan performa model yang baik dalam melakukan prediksi.

**Kata kunci:** Klasifikasi, Pohon Keputusan, Python, Google Collaboration

## LATAR BELAKANG

Di era digital saat ini, kemajuan teknologi informasi mengalami pertumbuhan yang pesat. Salah satu perkembangan yang signifikan terjadi dalam bidang kecerdasan buatan (*Artificial Intelligence*, AI), yang telah diterapkan luas di berbagai sektor, termasuk analisis data dan pengambilan keputusan. Para peneliti di bidang data *mining* telah menciptakan beragam algoritma klasifikasi yang meningkatkan proses klasifikasi dengan memanfaatkan atribut numerik dan nominal. Penerapan data *mining* dalam sektor pendidikan, yang dikenal sebagai *Educational Data Mining* (EDM), bertujuan untuk mengembangkan metode yang dapat mengekstrak pengetahuan berharga dari data dalam konteks pendidikan (Syahara et al., 2021).

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).

*Decision Tree* digunakan untuk menganalisis klasifikasi dan pola prediksi data serta menggambarkan hubungan antara variabel atribut  $x$  dan variabel target  $y$  dalam bentuk struktur pohon (Syahara et al., 2021). *Decision Tree* merupakan struktur yang menyerupai *flowchart*, di mana setiap *node* internal melakukan pengujian terhadap variabel atribut, dan cabangnya mencerminkan hasil pengujian tersebut, sedangkan *node* terluar (*leaf*) berfungsi sebagai label (Dzikrina, 2021). Penerapan *Decision Tree* dalam klasifikasi data sangat relevan, khususnya dalam konteks evaluasi mobil. Penilaian mobil memiliki peran penting bagi konsumen dan produsen, dan dengan menggunakan algoritma *Decision Tree*, dapat dibuat model untuk mengklasifikasikan mobil berdasarkan fitur atau atribut yang dimilikinya.

Python sebagai bahasa pemrograman populer dalam pengembangan kecerdasan buatan, menyediakan pustaka dan *framework* yang mendukung implementasi algoritma *Decision Tree*. Salah satu pustaka umum yang digunakan adalah *scikit-learn*, yang menawarkan implementasi efisien dan mudah digunakan untuk berbagai algoritma *machine learning*, termasuk *Decision Tree*. Dengan menerapkan algoritma *Decision Tree* untuk klasifikasi data evaluasi mobil menggunakan Python, kita dapat memanfaatkan kekuatan AI untuk memberikan solusi efektif dan efisien dalam pengambilan keputusan terkait evaluasi mobil. Hal ini dapat membantu konsumen membuat keputusan pembelian yang lebih informatif dan mendukung produsen dalam meningkatkan kualitas produk mereka.

## KAJIAN TEORITIS

### Data Mining

Metode yang dipakai untuk menggali informasi yang belum ditemukan dengan cara manual dari suatu kumpulan data disebut dengan data *mining* (Merawati & Rino, 2019). Kata data *mining* mulai dikenal hal ini karena kebutuhan untuk mengolah data adalah hal berguna dan sangat penting dilakukan dalam berbagai bidang keilmuan, kesehatan, pendidikan dan bahkan industri. Ilmu data *mining* merupakan perpaduan ilmu dari *artificial intelligence*, statistik, dan penelitian basis data yang selalu meningkat. Pola ini akan dikenali dan dipelajari dengan teliti oleh perangkat yang diharapkan nantinya dapat memberikan analisa yang berguna. Data mining merupakan proses pembelajaran komputer (*machine learning*) dengan menggunakan satu atau lebih teknik pembelajaran (algoritma), sehingga diharapkan komputer mampu menganalisa pengetahuan secara otomatis. Dalam data *mining* terdapat 6 metode yang biasa di jalankan yaitu ramalan atau prediksi, penggambaran atau deskripsi, klasifikasi, estimasi, asosiasi dan *clustering*.

### Klasifikasi

Klasifikasi adalah proses pencarian sekumpulan model, pola atau fungsi yang menggambarkan dan membedakan objek data untuk dikelompokkan ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Pada klasifikasi proses mencari karakteristik sebuah objek dilakukan, selanjutnya objek dengan karakteristik yang sama dimasukkan ke dalam salah satu kelas yang sudah diartikan terlebih dahulu (Hafizan & Putri, 2020). Dalam klasifikasi dataset yang digunakan harus memiliki label atau atribut tujuan. Akurasi klasifikasi dihitung dengan menentukan persentase instance yang diberi label kelas yang benar. Tujuan klasifikasi untuk dapat memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui (Putra & Chan, 2018).

### Decision Tree

*Decision Tree* adalah struktur flowcart yang mempunyai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes pada atribut. Setiap cabang merepresentasikan kelas atau distribusi kelas, alur pada *decision tree* ditelusuri dari simpul ke akar ke simpul daun yang memegang prediksi kelas untuk contoh tersebut. *Decision tree* mudah untuk dikonversikan keaturan klasifikasi (*classification rule*). Konsep data dalam *decision tree* dinyatakan dalam bentuk tabel dengan atribut dan *record* (Wardhana et al., 2021).

Dalam pembangunan *Decision Tree* langkah awalnya melibatkan evaluasi semua atribut yang tersedia menggunakan suatu ukuran statistik, seperti *information gain*, untuk menilai sejauh mana suatu atribut efektif dalam mengelompokkan sampel data. Atribut yang dipilih

untuk ditempatkan di *root\_node* adalah atribut yang memiliki *information gain* tertinggi. Semua atribut ini bersifat kategori dengan nilai diskrit, dan apabila terdapat atribut dengan nilai kontinu, perlu dilakukan proses diskritisasi. Untuk mencari nilai gain (*information Gain*) dapat dilakukan dengan konsep *Entropy*, *Gini Index* dan *Classification Error*.

- *Information Gain* dengan Nilai *Entropy*

$$Entropy(S) = \sum_{i=1}^n p(i|s) \log_2 p(i|s)$$

Dimana:

$S$  : Himpunan kasus dalam dataset

$n$  : Jumlah partisi atribut  $S$

$P_i$  : proporsi dari  $S_i$  terhadap  $S$

Nilai *entropy* ini akan digunakan untuk menghitung nilai *gain*. Adapun rumus untuk mencari *gain* adalah sebagai berikut:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy S_i$$

$S$  : Himpunan kasus

$A$  : Atribut

$n$  : Jumlah partisi pada atribut  $A$

$|S_i|$  : Jumlah kasus pada partisi ke 1

$|S|$  : Jumlah kasus dalam  $S$

- *Information Gain* dengan nilai *Gini Index*

$$Gini(t) = 1 - \sum_{i=1}^n [p(i|s)]^2$$

$$Gain(S, A) = Gini(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} Gini(S_i)$$

- *Information Gain* dengan *Classification Error* nilai  $C$ . Error diperoleh dari nilai *value* atribut yang terkecil dari terhadap *class label*.

$$C. Error(S) = 1 - \max_i [p(i|s)]$$

$$Gain(S, A) = Error(s) - \sum_{i=1}^n \frac{|S_i|}{|S|} Error(S_i)$$

## **Python**

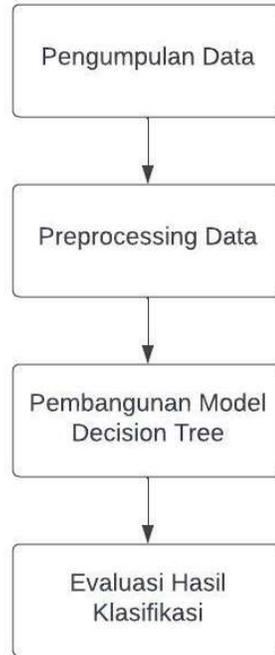
Python adalah scripting language yang berorientasi objek. Bahasa pemrograman ini dapat digunakan untuk pengembangan perangkat lunak dan bisa dijalankan melalui berbagai sistem operasi. Saat ini, Python juga merupakan bahasa yang populer bagi bidang data *science* dan analisis. Hal ini dikarenakan oleh dukungan bahasa Python terhadap *library* yang didalamnya menyediakan fungsi analisis data dan fungsi *machine learning*, data *preprocessing tools*, serta visualisasi data (Orvin, 2018).

Proses implementasi *desicion tree* dengan python dapat menggunakan berbagai editor, salah satunya adalah dengan menggunakan *google colab*. *Google Colab* merupakan *platform* pengembangan berbasis *cloud* yang memungkinkan pengguna untuk melakukan pemrograman dan analisis data (Diariono et al., 2022).

- 1) Pengembangan program dilakukan dengan cepat dan coding yang lebih sedikit
- 2) Mendukung multi-platform
- 3) Memiliki sistem pengelolaan memori yang otomatis
- 4) Python bersifat *Object Oriented Programming* (OOP)

## **METODE PENELITIAN**

Penelitian dilakukan menggunakan metode penyajian data secara deskriptif kuantitatif. Deskriptif kuantitatif adalah sebuah jenis penelitian yang digunakan untuk menganalisis data dengan cara mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya, dan menarik kesimpulan dari fenomena yang dapat diamati dengan menggunakan angka-angka. *Decision Tree* adalah salah satu algoritma *machine learning* yang digunakan untuk pengambilan sebuah keputusan berbasis aturan. *Decision Tree* dikenal juga sebagai model prediktif yang dapat digunakan untuk klasifikasi dan regresi. Rincian langkah-langkah penelitian dapat ditemukan dalam *flowchart* dibawah.



**Gambar 1. Flowchart Langkah-langkah penelitian**

## HASIL DAN PEMBAHASAN

### Pengumpulan Data

Pada bagian ini memuat proses pengumpulan data. *Dataset* yang digunakan dalam penelitian ini merupakan informasi mengenai evaluasi mobil yang diambil dari sumber terbuka, yaitu <https://archive.ics.uci.edu/> atau data dari *machine learning* yang mempunyai judul *Car Evaluation*. *Dataset* ini mencakup 7 atribut kelas, sebagaimana tercantum pada Tabel 1.

**Tabel 1. Atribut Dataset**

<i>Atribut</i>	<i>Keterangan</i>
<i>Buying</i>	Harga mobil
<i>Maint</i>	Harga perawatan mobil
<i>Doors</i>	Jumlah pintu mobil
<i>Person</i>	Kapasitas penumpang mobil
<i>Lug_boot</i>	Ukuran bagasi mobil
<i>Safety</i>	Estimasi keamanan mobil
<i>class</i>	Tingkat evaluasi ( <i>unacceptable, acceptable, good, very good</i> )

## **Preprocessing Data**

Pada tahapan ini memiliki beberapa tahapan lagi seperti, mengimpor *dataset* yang memungkinkan peneliti untuk mengupload data dari sumber eksternal untuk implementasikan ke program. Selanjutnya ada tahapan manipulasi data yaitu proses seperti penghapusan baris dengan nilai tertentu, mengatasi *missing value*, melakukan deklarasi, serta melakukan split data. Selanjutnya yaitu melakukan visualisasi terhadap data agar memudahkan pemahaman pada pola data.

## **Pembangunan Model Decision Tree**

Pada penelitian ini dibuat dua buah model klasifikasi *decision tree*, yaitu sebagai berikut.

### **1. Klasifikasi Decision Tree dengan Kriteria Indeks Gini**

Klasifikasi *Decision Tree* dengan kriteria indeks gini memiliki beberapa tahapan yaitu sebagai berikut.

#### **a) Membuat model Klasifikasi Decision Tree dengan kriteria indeks gini**

Tahapan pertama yaitu pembuatan model klasifikasi *Decision Tree* dengan menggunakan kriteria indeks gini, tahapan ini dilakukan dengan dengan mengimpor *package* klasifikasi *Decision Tree* dari *sklern tree*. Setelah itu dibuat sebuah *instance* model klasifikasi *Decision Tree* dengan kriteria indeks gini dan dilakukan pencocokan dengan *training* data. Selanjutnya yaitu melakukan prediksi hasil data *test* dengan kriteria indeks gini.

#### **b) Cek skor akurasi dengan kriteria indeks gini**

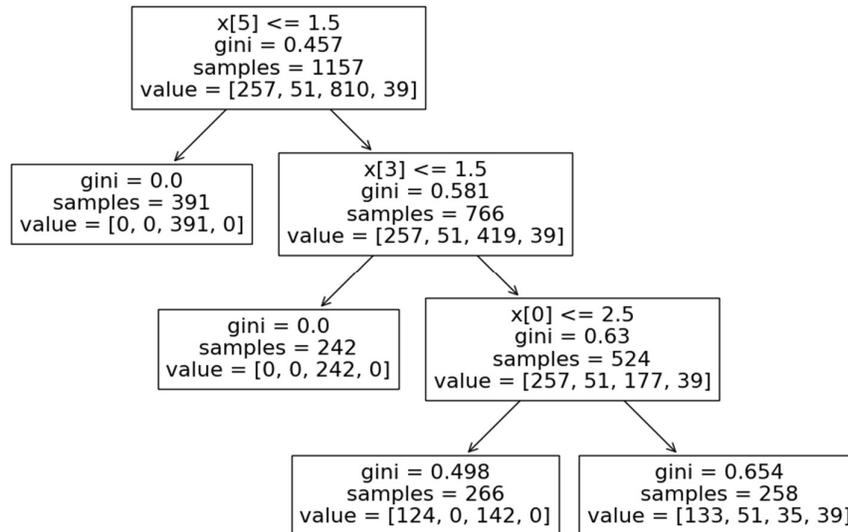
Dengan model klasifikasi *Decision Tree* dengan kriteria indeks gini diperoleh skor akurasi dengan menggunakan data *training* adalah sebesar 78%, serta skor akurasi dengan menggunakan data *testing* adalah sebesar 80%.

#### **c) Pemeriksaan Overfitting dan Underfitting**

*Overfitting* adalah kondisi dimana nilai akurasi pada data *training* tinggi dengan nilai akurasi data *testing* rendah. Sedangkan *underfitting* merupakan kondisi saat nilai akurasi data *training* rendah diikuti dengan nilai akurasi data *testing* yang rendah. Pada model klasifikasi *Decision Tree* dengan menggunakan kriteria indeks gini tidak ada tanda-tanda *overfitting* dan *underfitting*, dikarenakan kedua nilai skor akurasi cukup sebanding.

#### d) Visualisasikan Decision Tree

Tahapan selanjutnya yaitu melakukan visualisasi *Decision Tree* dengan kriteria indeks gini



**Gambar 2. Visualisasi Decision Tree dengan kriteria indeks gini**

## 2. Klasifikasi Decision Tree dengan Kriteria Entropy

Klasifikasi *Decision Tree* dengan kriteria *entropy* memiliki beberapa tahapan yaitu sebagai berikut.

### a) Membuat model Klasifikasi Decision Tree dengan kriteria entropy

Pada tahapan dibuat model klasifikasi *Decision Tree* dengan menggunakan kriteria *entropy*, tahapan ini dilakukan dengan dengan mengimpor *package* klasifikasi *Decision Tree* dari *sklearn tree*. Setelah itu dibuat sebuah *instance* model klasifikasi *Decision Tree* dengan kriteria *entropy* dan dilakukan pencocokan dengan training data. Selanjutnya yaitu melakukan prediksi hasil data *testing* dengan kriteria *entropy*.

### b) Cek skor akurasi dengan kriteria entropy

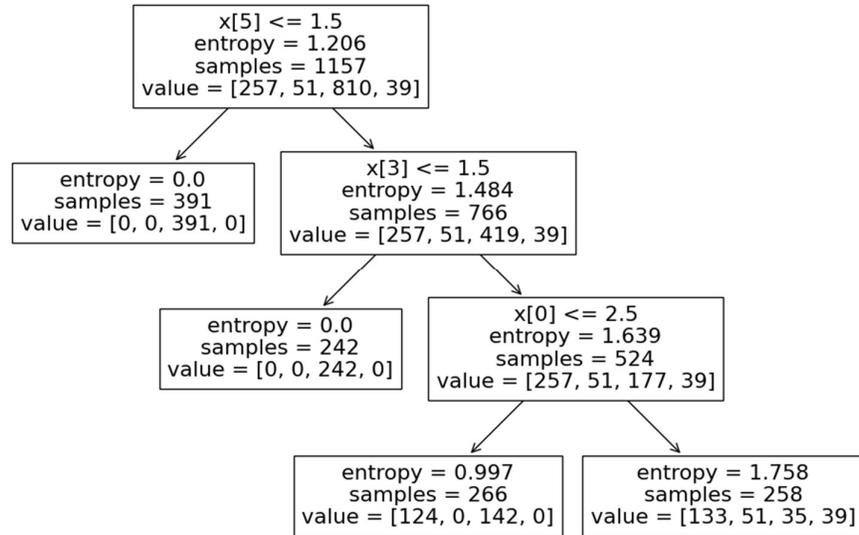
Dengan model klasifikasi *Decision Tree* dengan kriteria *entropy* diperoleh skor akurasi yang sama dengan model *Decision Tree* dengan kriteria indeks gini yaitu dengan menggunakan data training adalah sebesar 78%, serta skor akurasi dengan menggunakan data *testing* adalah sebesar 80%.

### c) Pemeriksaan Overfitting dan Underfitting

Dikarenakan skor akurasi pada model klasifikasi *Decision Tree* dengan menggunakan kriteria *entropy* sama dengan model *Decision Tree* dengan kriteria indeks gini yang memiliki skor akurasi yang sebanding atau tidak beda jauh antara skor akurasi data *training* dengan skor akurasi data *testing*, maka tidak ada tanda-tanda *overfitting* dan *underfitting*.

### e) Visualisasikan Decision Tree

Tahapan selanjutnya yaitu melakukan visualisasi *Decision Tree* dengan kriteria *entropy*.



**Gambar 3. Visualisasi Decision Tree dengan kriteria entropy**

### Evaluasi Hasil Klasifikasi

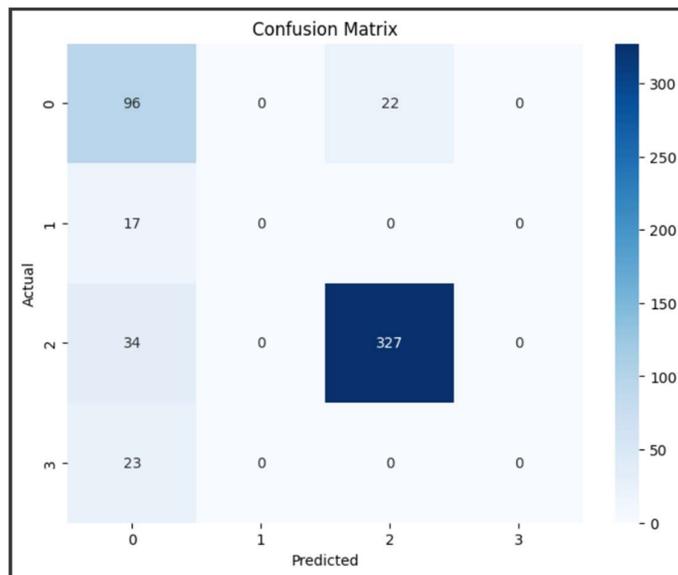
Berikut ini adalah table dari hasil akhir dari model klasifikasi *Decision tree* yang menggunakan data *Car Evaluation*.

**Tabel 2. Hasil Klasifikasi Decision Tree**

	precision	recall	f1-score	support
acc	0.56	0.56	0.56	127
good	0.00	0.00	0.00	18
unacc	0.87	0.97	0.92	399
vgood	0.00	0.00	0.00	26
accuracy			0.81	570
macro avg	0.36	0.38	0.37	570
weighted avg	0.74	0.81	0.77	570

Hasil kasifikasi dengan model *Decision Tree* untuk data *Evaluation Car* menunjukkan *accuracy* sebesar 81%, dimana model ini menunjukkan kemampuan yang baik dalam memberikan prediksi yang sesuai dengan data *testing*. *Precision* atau ketepatan model dalam

mengidentifikasi kelas setiap atribut menunjukkan nilai sebesar 56% untuk “*acceptable*”, 0% untuk “*good*”, 97% untuk “*unacceptable*” serta 0% untuk “*very good*”. Untuk *recall* atau kemampuan model dalam mendeteksi *instance* dari kelas tertentu, dimana diperoleh nilai sebesar 56% untuk “*acceptable*”, 0% untuk “*good*”, 87% untuk “*unacceptable*” serta 0% untuk “*very good*”. Serta *F1-Score* atau nilai *Harmonic Mean* (Rata-rata Harmonik) dari *Precision* dan *Recall*, dimana diperoleh nilai sebesar 56% untuk “*acceptable*”, 0% untuk “*good*”, 92% untuk “*unacceptable*” serta 0% untuk “*very good*”.



**Gambar 4. Confusion Matrix**

Pada *Confusion Matrix* merincikan hasil dari klasifikasi dengan model *Decision Tree* dengan menunjukkan *instance* yang prediksinya benar atau salah. Dimana ditunjukkan bahwa prediksi yang benar untuk “*acceptable*” sebanyak 96, “*good*” sebanyak 0, “*unacceptable*” sebanyak 327, dan “*very good*” sebanyak 0. Untuk prediksi yang salah untuk “*acceptable*” sebanyak 22, “*good*” sebanyak 17, “*unacceptable*” sebanyak 34, dan “*very good*” sebanyak 23.

## KESIMPULAN DAN SARAN

Dalam proyek ini, dibuat model Pengklasifikasi *Decision Tree* untuk memprediksi kelayakan mobil. Dibangun dua buah model yaitu, menggunakan kriteria indeks gini dan satu lagi menggunakan kriteria *entropy*. Model ini menghasilkan kinerja yang sangat baik yang ditunjukkan oleh akurasi model pada kedua kasus yaitu 0,8021. Pada model dengan kriteria indeks gini, skor akurasi data *training* sebesar 0,7865 sedangkan akurasi data *testing* sebesar 0,8021. Kedua nilai ini cukup sebanding. Jadi, tidak ada tanda-tanda *overfitting* dan *underfitting*. Demikian pula pada model dengan kriteria *entropy*, skor akurasi *training* adalah

0,7865 sedangkan akurasi *testing* menjadi 0,8021. Kita mendapatkan nilai yang sama seperti pada kasus dengan kriteria gini. Jadi, tidak ada tanda-tanda *overfitting* dan *underfitting*. Dalam kedua kasus tersebut, skor akurasi *training* dan *testing* adalah sama. Ini mungkin terjadi karena kumpulan data yang kecil. *Confusion Matrix* dan laporan klasifikasi menghasilkan performa model yang sangat baik.

## DAFTAR REFERENSI

- Diartono, D. A., Zuliarso, E., Sulastri, S., & Anis, Y. (2022). PELATIHAN PEMROGRAMAN BERBASIS KOMPUTASI AWAN UNTUK Mendukung Pembelajaran Jarak Jauh Bagi Siswa SMK Negeri 4 Kendal. *Jurnal Pengabdian Masyarakat Intimas (Jurnal INTIMAS): Inovasi Teknologi Informasi Dan Komputer Untuk Masyarakat*, 2(1), 1–6. <https://doi.org/10.35315/intimas.v2i1.8841>
- Dzikrina, I. (2021). Penerapan Data Mining untuk Prediksi Penyakit. *Informatika*, 20(2), 155–159. <http://repository.uin-suska.ac.id/23008/%0Ahttp://repository.uin-suska.ac.id/23008/2/NURFITRIANTI.pdf>
- Hafizan, H., & Putri, A. N. (2020). Penerapan Metode Klasifikasi Decision Tree Pada Status Gizi Balita Di Kabupaten Simalungun. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, 1(2), 68–72. <https://doi.org/10.30645/kesatria.v1i2.23>
- Hana, F. M. (2020). Klasifikasi Penderita Penyakit Diabetes Menggunakan Algoritma Decision Tree C4.5. *Jurnal SISKOM-KB (Sistem Komputer Dan Kecerdasan Buatan)*, 4(1), 32–39. <https://doi.org/10.47970/siskom-kb.v4i1.173>
- Maulida, L. (2018). Penerapan Datamining Dalam Mengelompokkan Kunjungan Wisatawan Ke Objek Wisata Unggulan Di Prov. Dki Jakarta Dengan K-Means. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 2(3), 167. <https://doi.org/10.14421/jiska.2018.23-06>
- Merawati, D., & Rino. (2019). Penerapan data mining penentu minat Dan bakat siswa Smk dengan metode C4 . 5. In *Jurnal Algor* (Vol. 1, Issue 1, pp. 28–37).
- Orvin, E. T. (2018). *Penggunaan Python untuk Data Mining*. Binus University School of Computer Science.
- Putra, P. P., & Chan, A. S. (2018). Pengembangan Aplikasi Perhitungan Prediksi Stock Motor Menggunakan Algoritma C 4.5 Sebagai Bagian dari Sistem Pengambilan Keputusan (Studi Kasus di Saudara Motor). *INOVTEK Polbeng - Seri Informatika*, 3(1), 24. <https://doi.org/10.35314/isi.v3i1.296>
- Syahara, Z., Adiha, R. N., & Windarto, A. P. (2021). Implementasi Data Mining Algoritma Apriori Pada Sistem Persediaan Bahan Bangunan Di Karang Sari. *Brahmana : Jurnal Penerapan Kecerdasan Buatan*, 2(2), 107–115. <https://doi.org/10.30645/brahmana.v2i2.72>
- Wardhana, A. W., Patimah, E., Shafarindu, A. I., Siahaan, Y. M., Haekal, B. V., & Prasvita, D. S. (2021). Klasifikasi Data Penjualan pada Supermarket dengan Metode Decision Tree. *Senamika*, 2(1), 660–667. <https://conference.upnvj.ac.id/index.php/senamika/article/view/1389>