

Analisis Perbandingan Kinerja Model *K-Nearest Neighbors* dan *Decision Tree* untuk Prediksi Pengeluaran Nasabah

Shindy Yuliyatini^{1*)}, Via Olga Pangaribuan²⁾ Adnan Nuur Bachtiar³⁾

¹Program Studi Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur

^{2,3}Program Studi Teknik Informatika, Fakultas Ilmu Rekayasa, Universitas Paramadina

Email: ¹2311601302@student.budiluhur.ac.id ²via.pangaribuan@students.paramadina.ac.id,

³adnan.bachtiar@students.paramadina.ac.id

Abstrak - Prediksi pengeluaran nasabah merupakan aspek krusial dalam analisis data keuangan guna membantu institusi perbankan memahami pola perilaku konsumen. Penelitian ini membandingkan kinerja dua algoritma pembelajaran mesin, yaitu *K-Nearest Neighbors* (KNN) dan *Decision Tree*, dalam memprediksi pengeluaran nasabah. Sampel yang digunakan berupa data transaksi seorang nasabah pada Bank BCA dengan total 2.567 transaksi. Evaluasi dilakukan menggunakan tiga metrik utama, yakni *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE). Hasil penelitian menunjukkan bahwa algoritma KNN menghasilkan tingkat kesalahan prediksi yang lebih rendah dibandingkan *Decision Tree* pada seluruh metrik evaluasi, sehingga dinilai lebih efektif dalam tugas prediksi ini.

Kata kunci: KNN, Decision Tree, Prediksi Pengeluaran, MAE, MSE

Abstract - Customer expenditure prediction is a crucial aspect of financial data analysis, helping banking institutions better understand consumer behavior. This study compares the performance of two machine learning algorithms, *K-Nearest Neighbors* (KNN) and *Decision Tree*, in predicting customer expenditures. The dataset used consists of 2,567 transaction records from a single customer at Bank BCA. The performance of both models is evaluated using three key metrics: *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), and *Root Mean Squared Error* (RMSE). The results show that the KNN algorithm outperforms the *Decision Tree* by producing lower prediction errors across all evaluation metrics, making it more effective for this predictive task.

Keywords— KNN, Decision Tree, Expense Prediction, MAE, MSE

I. PENDAHULUAN

Machine learning (pembelajaran mesin) merupakan cabang dari kecerdasan buatan yang memungkinkan sistem untuk belajar dari data dan membuat keputusan atau prediksi tanpa perlu diprogram secara eksplisit [1]. Dalam konteks keuangan, machine learning dapat diterapkan untuk menganalisis data nasabah, seperti informasi pribadi, riwayat transaksi, status kredit, dan aspek keuangan lainnya [2], [3]. Dengan menggunakan dataset yang besar dan beragam, machine learning dapat mengidentifikasi pola yang mungkin

sulit ditemukan oleh manusia, sehingga membantu lembaga keuangan dalam mengambil keputusan yang lebih tepat dan efisien [4], [5], [6].

Sebagai contoh, dalam analisis kelayakan kredit, lembaga keuangan sering kali mengandalkan dataset yang berisi informasi tentang pendapatan, utang, skor kredit, dan riwayat pembayaran nasabah untuk menentukan apakah nasabah tersebut layak mendapatkan pinjaman. Dengan *machine learning*, sistem dapat dilatih untuk mengenali pola dalam data tersebut dan memberikan prediksi mengenai kemampuan nasabah dalam membayar pinjaman di masa depan. Hal ini sangat berguna untuk mengurangi risiko dan meningkatkan akurasi dalam proses pemberian kredit.

Penelitian ini bertujuan untuk memprediksi besarnya pengeluaran nasabah dengan memanfaatkan teknik pembelajaran mesin. Dua algoritma yang dibandingkan dalam penelitian ini adalah *K-Nearest Neighbors* (KNN) [7] dan *Decision Tree* [8], [9], yang dipilih karena pendekatan dan karakteristik klasifikasinya yang berbeda. Data yang digunakan berupa riwayat transaksi keuangan dari seorang nasabah Bank BCA dengan total 2.567 entri transaksi, yang mencerminkan berbagai jenis pengeluaran dalam periode tertentu. Data ini kemudian dibersihkan dan diproses untuk menghasilkan fitur-fitur yang relevan sebelum diterapkan ke masing-masing model. Kinerja kedua algoritma dievaluasi menggunakan tiga metrik utama, yaitu *Mean Absolute Error* (MAE) [10], *Mean Squared Error* (MSE) [11], dan *Root Mean Squared Error* (RMSE) [12], untuk menilai seberapa akurat model dalam memprediksi nominal pengeluaran. Melalui pendekatan ini, penelitian diharapkan dapat memberikan gambaran mengenai algoritma yang lebih tepat digunakan dalam analisis perilaku keuangan nasabah berdasarkan riwayat transaksinya.

Dataset nasabah yang digunakan dalam penelitian ini memiliki karakteristik yang serupa dengan dataset perbankan pada umumnya, yang mencakup beragam informasi seperti usia, jenis kelamin, status pekerjaan, pendapatan, jumlah utang, serta catatan pembayaran tagihan. Data-data ini tidak hanya relevan untuk analisis pengeluaran, tetapi juga dapat dimanfaatkan dalam berbagai aplikasi machine learning lanjutan, seperti klasifikasi untuk menentukan kelayakan pemberian pinjaman, regresi untuk memprediksi jumlah

pinjaman yang sesuai, maupun segmentasi untuk mengelompokkan nasabah berdasarkan profil risiko.

Dengan kemajuan teknologi dan ketersediaan data yang semakin besar, penerapan machine learning dalam analisis keuangan nasabah menjadi semakin penting [13]. Dengan memanfaatkan pendekatan machine learning, penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan efisiensi operasional lembaga keuangan. Penggunaan model prediktif seperti KNN dan *Decision Tree* dapat membantu mengurangi ketergantungan pada analisis manual yang rentan terhadap kesalahan manusia, sekaligus memberikan pemahaman yang lebih mendalam mengenai pola perilaku keuangan nasabah. Pada akhirnya, hal ini dapat mendukung pengambilan keputusan yang lebih akurat, cepat, dan berbasis data dalam berbagai layanan perbankan.

II. METODE PENELITIAN

Metode penelitian yang digunakan dengan pendekatan kuantitatif dengan memanfaatkan teknik pembelajaran mesin dan analisis data yang sistematis.:

A. Desain Penelitian

1. Jenis Penelitian.

Penelitian ini merupakan penelitian analitis, yang bertujuan untuk menganalisis dan membandingkan kinerja dua model pembelajaran mesin, yaitu KNN dan *Decision Tree*, dalam memprediksi pengeluaran nasabah.

2. Pendekatan

Penelitian. Penelitian ini menggunakan pendekatan kuantitatif, di mana data yang dikumpulkan akan dianalisis menggunakan teknik pembelajaran mesin untuk membandingkan akurasi model KNN dan *Decision Tree* dalam prediksi pengeluaran nasabah.

3. Populasi.

Populasi penelitian ini adalah data pengeluaran nasabah yang diperoleh dari transaksi bank. Data tersebut mencakup variabel-variabel seperti jumlah pengeluaran, jumlah yang diterima, loan, dan lain-lain.

4. Sampel.

Sampel yang digunakan adalah dataset dari transaksi seorang nasabah pada bank BCA dengan ukuran sampel 2567 transaksi / sampel. Data tersebut akan dipilih secara acak, dengan memperhatikan representasi yang seimbang antara berbagai kategori yang relevan.

B. Prosedur Penelitian

Penelitian ini berfokus pada pengembangan dan pengujian dua model pembelajaran mesin. Dengan model

a) KNN

K-Nearest Neighbor (KNN) adalah algoritma dalam machine learning yang sering digunakan untuk klasifikasi dan regresi, yang bekerja dengan cara mencari data yang paling mirip (tetangga terdekat) dengan data yang ingin diprediksi atau diklasifikasikan [14]. Konsep dasar dari model KNN adalah K yang merujuk pada jumlah "tetangga" terdekat yang dipertimbangkan untuk memutuskan kelas atau nilai *output* dari suatu data baru. *Nearest Neighbors* adalah KNN bekerja dengan menghitung jarak antara data baru yang akan diprediksi dan data yang ada di dalam dataset. Jarak ini umumnya dihitung dengan menggunakan beberapa metode

pengukuran jarak, seperti *Euclidean*, *Manhattan*, atau *Minkowski*.

b) Decision Tree

Decision Tree adalah salah satu algoritma dalam machine learning yang digunakan untuk melakukan klasifikasi dan regresi [15]. Model ini menggambarkan keputusan dan kemungkinan hasilnya dalam bentuk pohon terstruktur, di mana setiap simpul (*node*) menggambarkan kondisi atau pertanyaan yang perlu dijawab untuk mengklasifikasikan data, dan setiap cabang menunjukkan hasil dari keputusan tersebut.

Komponen utama dalam *Decision Tree*

- *Root Node*. Titik awal pohon yang mewakili seluruh dataset. Di sini, algoritma memilih fitur yang terbaik untuk memulai pembagian data.
- *Decision Nodes*. Titik di dalam pohon tempat pembagian lebih lanjut terjadi berdasarkan fitur-fitur dalam dataset.
- *Leaf Nodes*. Titik akhir dalam pohon yang menunjukkan keputusan akhir atau kelas yang diprediksi (untuk klasifikasi) atau nilai yang diprediksi (untuk regresi).
- *Edges*. Cabang yang menghubungkan simpul-simpul dalam pohon, menunjukkan bagaimana data dibagi berdasarkan nilai fitur.

Proses pengujian dilakukan dengan membagi dataset menjadi data latih dan data uji menggunakan metode *train-test split* untuk memastikan bahwa model dapat diuji pada data yang belum pernah dilihat sebelumnya. Kedua algoritma, yaitu *K-Nearest Neighbors* (KNN) dan *Decision Tree*, dilatih menggunakan data latih, lalu diuji performanya pada data uji. Evaluasi dilakukan berdasarkan tiga metrik utama, yaitu *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE), yang mengukur seberapa besar rata-rata kesalahan prediksi yang dihasilkan oleh masing-masing model. Hasil pengujian dari kedua model kemudian dibandingkan untuk menentukan metode yang paling efektif dalam memprediksi pengeluaran nasabah berdasarkan pola data yang tersedia.

III. HASIL DAN DISKUSI

A. Model KNN

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# Load dataset (ganti path dengan lokasi dataset Anda)
dataset_path = "UAS_Dataset_bank_BCA.csv"
data = pd.read_csv(dataset_path, delimiter=';')

data.info() # Periksa tipe data
data.head()
```

Gambar 1. *Import library* KNN dan dataset

Untuk menyelesaikan penelitian ini, langkah pertama yang penulis lakukan adalah mengimport librari dan juga dataset yang akan digunakan. Berdasarkan Gambar 1, beberapa *library* yang penulis

- a. Pandas = untuk manipulasi data.
- b. `sklearn.model_selection.train_test_split` = untuk membagi dataset menjadi data latih dan uji.
- c. `sklearn.neighbors.KNeighborsRegressor` = model KNN untuk regresi.
- d. `sklearn.metrics` = untuk menghitung metrik evaluasi seperti *Mean Absolute Error* (MAE) dan *Mean Squared Error* (MSE).
- e. Numpy = untuk operasi numerik.

SL No.	Bank Name	Account Number	Date	Cheque No.	Original Amount	Original Balance	Category	Converted Amount	Converted Balance	Conversion rate	
0	2051	Bank Central Asia (BCA) Indonesia	30-Apr-23	Hasabah B	NaN	-24310396.00	340.841.663.42	Loan	-24310396.00	340.841.663.42	1.00
1	2052	Bank Central Asia (BCA) Indonesia	30-Apr-23	Hasabah B	NaN	10.000.000.00	350.841.663.42	Transfer from	10.000.000.00	350.841.663.42	1.00
2	2053	Bank Central Asia (BCA) Indonesia	30-Apr-23	Hasabah B	NaN	1.150.000.00	351.991.663.42	Transfer from	1.150.000.00	351.991.663.42	1.00
3	2054	Bank Central Asia (BCA) Indonesia	30-Apr-23	Hasabah B	NaN	6.86	351.991.670.28	Interest Connection	6.86	351.991.670.28	1.00
4	2055	Bank Central Asia (BCA) Indonesia	30-Apr-23	Hasabah B	NaN	1.156.30	351.992.826.58	Interest	1.156.30	351.992.826.58	1.00

Gambar 2. Hasil pembacaan dataset model KNN

Pada Gambar 2, terlihat hasil dataset yang telah di *import*. Selanjutnya, dilakukan pebersihan dan pemformatan data sehingga data yang ada siap untuk diterapkan dalam *machine learning* seperti pada Gambar 3. Beberapa hal yang penulis lakukan adalah:

- a. Kolom dengan nilai data numerik seperti original amount, original balance penulis ubah dan sesuaikan.
- b. Menghapus pemisah ribuan titik (.)
- c. Mengganti koma (*decimal*) dengan titik (.)
- d. Mengonversi nilai string menjadi tipe numerik menggunakan `pd.to_numeric`

```
# Membersihkan dan memformat data
numeric_columns = ['Original Amount', 'Original Balance', 'Converted Amount', 'Converted Balance', 'Conversion rate']
for col in numeric_columns:
    data[col] = data[col].str.replace('.', '', regex=False) # Hapus pemisah ribuan
    data[col] = data[col].str.replace(',', '.', regex=False) # Ganti koma dengan titik
    data[col] = pd.to_numeric(data[col], errors='coerce') # Konversi ke numerik

data.info()
data.head()
```

Gambar 3. Pra-proses data untuk model KNN

```
# Feature engineering untuk kolom 'Date'
data['Date'] = pd.to_datetime(data['Date'], format='%d-%b-%y', errors='coerce')
data['Year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
data['Day'] = data['Date'].dt.day

data.head()
```

Gambar 4. Penyesuaian kolom waktu untuk model KNN

Pada Gambar 4, langkah yang dilakukan adalah pemformatan kolom *date* menjadi tipe data standar, kemudian membuat kolom baru yang digunakan untuk menampung

data *date* menjadi format terpisah yaitu *year*, *month*, *day/date*.

```
# Memilih fitur dan target
features = ['Original Balance', 'Converted Balance', 'Year', 'Month', 'Day']
X = data[features]
y = data['Original Amount']

# Menghapus baris dengan nilai NaN
X = X.dropna()
y = y.loc[X.index]
# menghapus kolom dengan nilai NaN
data = data.drop(columns=['Cheque No.'])

data.head()
```

Gambar 5. Memilih fitur untuk model KNN

Langkah pada Gambar 5 yang penulis lakukan adalah memilih fitur dan target yang akan digunakan. Fitur (x) merupakan kolom-kolom yang menjadi input untuk prediksi, sementara itu, Target (y) adalah kolom yang ingin diprediksi (pengeluaran nasabah).

```
# Membagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 6. Data latih dan data uji model KNN

Berdasarkan Gambar 6, penulis membagi dataset menjadi data latih dan uji dengan rasio tertentu, dengan 20% untuk data pengujian. Data latih digunakan untuk melatih model, dan data uji untuk mengevaluasi performa model. *Syntax* `random_state=42` berfungsi sebagai penjamin pembagian data yang sama setiap kali kode dijalankan.

```
# Membuat model K-Nearest Neighbors (KNN)
knn_model = KNeighborsRegressor(n_neighbors=5)
knn_model.fit(X_train, y_train)

# Melakukan prediksi
y_pred = knn_model.predict(X_test)
```

Gambar 7. Penerapan model KNN

Pada Gambar 7, proses pelatihan dilakukan menggunakan model KNN dengan menetapkan jumlah tetangga terdekat (k) sebagai parameter utama. Model kemudian dilatih menggunakan data latih yang telah dipersiapkan sebelumnya, dan selanjutnya digunakan untuk melakukan prediksi terhadap data uji. Proses ini bertujuan untuk mengamati bagaimana pemilihan nilai k memengaruhi akurasi prediksi pengeluaran nasabah.



```
# Evaluasi model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)

Mean Absolute Error (MAE): 24599637.220486406
Mean Squared Error (MSE): 3261509135390835.0
Root Mean Squared Error (RMSE): 57109623.84214096
```

Gambar 8. Evaluasi model KNN

Langkah pada Gambar 8, performa model dievaluasi menggunakan metrik:

- Mean Absolute Error (MAE) dimana rata-rata kesalahan absolut antara nilai sebenarnya dan prediksi diukur.
- Mean Squared Error (MSE) dimana rata-rata kuadrat kesalahan juga diukur.
- Root Mean Squared Error (RMSE) dimana akar dari MSE juga diukur, sering juga digunakan untuk menghasilkan interpretasi yang lebih mudah.

```
new_data = [[5000000, 205000000, 5200000, 24000000, 1.03]] # Contoh data baru
prediction = knn_model.predict(new_data)
print("Prediksi pengeluaran:", prediction)

Prediksi pengeluaran: [17252628.]
```

Gambar 9. Pengujian dengan data baru pada model KNN

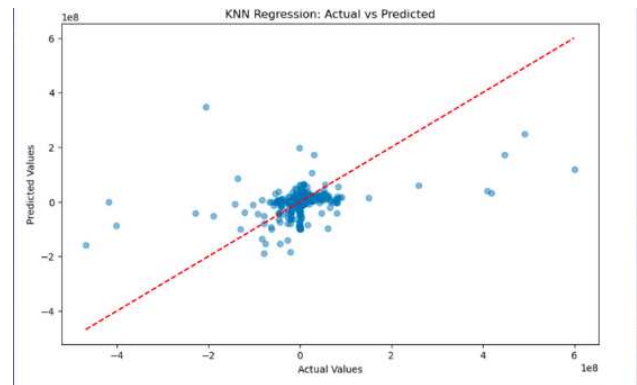
Pada Gambar 9, setelah pola prediksi diperoleh dari hasil pelatihan dan evaluasi model, langkah selanjutnya adalah melakukan pengujian terhadap data baru yang belum pernah digunakan sebelumnya. Data baru ini dimasukkan ke dalam model untuk melihat seberapa baik model dapat memprediksi pengeluaran berdasarkan pola yang telah dipelajari. Pengujian ini penting untuk mengukur kemampuan generalisasi model dalam menghadapi data dunia nyata. Hasil prediksi pada data baru kemudian dianalisis untuk memastikan bahwa performa model tetap konsisten dan dapat diandalkan dalam konteks penggunaan aktual di lingkungan perbankan.

```
# Visualisasi hasil prediksi vs nilai aktual (opsional)
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--r')
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("KNN Regression: Actual vs Predicted")
plt.savefig("knn_regression_actual_vs_predicted.png")
plt.show()
```

Gambar 10. Visualisasi hasil prediksi dengan KNN Regression

Untuk memahami sejauh mana akurasi model dalam melakukan prediksi, dilakukan visualisasi hasil prediksi dibandingkan dengan nilai aktual menggunakan model *K-Nearest Neighbors Regression* seperti pada Gambar 10.

Visualisasi ini ditampilkan pada Gambar 9, yang menunjukkan hubungan antara nilai pengeluaran yang diprediksi oleh model dan nilai pengeluaran aktual dari data uji. Grafik ini memberikan gambaran sejauh mana model mampu mengikuti pola data sebenarnya, serta membantu mengidentifikasi kemungkinan deviasi atau kesalahan prediksi yang terjadi. Hasil grafiknya seperti pada Gambar 11.



Gambar 11. Grafik visualisasi model KNN

B. Decision Tree

Langkah selanjutnya yang penulis lakukan adalah melakukan pengujian menggunakan model *Decision Tree*. Langkah yang sama juga dilakukan seperti pada model KNN yaitu melakukan import dimana *library sklearn.tree* adalah melakukan import *DecisionTreeRegressor*

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# load dataset (ganti path dengan lokasi dataset Anda)
dataset_path = "UAS_Dataset_bank_BCA.csv"
data = pd.read_csv(dataset_path, delimiter=',')

data.info() # Periksa tipe data
data.head()
```

Gambar 12. Import library DecisionTree dan dataset

Sebelum model *Decision Tree* diterapkan, langkah awal yang dilakukan adalah menampilkan tipe data dari setiap kolom dalam dataset untuk memastikan bahwa seluruh fitur memiliki format data yang sesuai dengan kebutuhan model. Pemeriksaan ini penting agar tidak terjadi kesalahan saat proses *training* atau prediksi. Selanjutnya, ditampilkan sebagian data dari dataset yang akan digunakan sebagai data uji seperti pada Gambar 13, guna memberikan gambaran mengenai struktur dan isi data yang digunakan dalam proses pengujian. Langkah ini juga membantu dalam memastikan bahwa data yang dimasukkan ke dalam model telah melalui proses pra-proses yang memadai dan siap untuk dianalisis lebih lanjut menggunakan algoritma *Decision Tree Regression*.



```
class pandas.core.frame.DataFrame
RangeIndex: 4799 entries, 0 to 4798
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  --
 0   Sl. No.     4799 non-null   int64
 1   Bank Name   4799 non-null   object
 2   Account Number  4799 non-null  object
 3   Date       4799 non-null   object
 4   Cheque No.  4799 non-null   float64
 5   Original Amount  4799 non-null float64
 6   Original Balance  4799 non-null float64
 7   Category    4799 non-null   object
 8   Converted Amount  4799 non-null float64
 9   Converted Balance  4799 non-null float64
10   Conversion rate  4799 non-null float64
dtypes: float64(9), int64(1), object(1)
memory usage: 412.5+ KB
```

Sl. No.	Bank Name	Account Number	Date	Cheque No.	Original Amount	Original Balance	Category	Converted Amount	Converted Balance	Conversion rate	
0	2551	Bank Central Asia (BCA) Indonesia	Nasabah B	30-Apr-23	NaN	-24510996.00	3.408417e+08	Loan	-24510996.00	3.408417e+08	1.0
1	2552	Bank Central Asia (BCA) Indonesia	Nasabah B	30-Apr-23	NaN	10000000.00	3.508417e+08	Transfer from	10000000.00	3.508417e+08	1.0
2	2553	Bank Central Asia (BCA) Indonesia	Nasabah B	30-Apr-23	NaN	1150000.00	3.519917e+08	Transfer from	1150000.00	3.519917e+08	1.0
3	2554	Bank Central Asia (BCA) Indonesia	Nasabah B	30-Apr-23	NaN	6.80	3.519917e+08	Interest Connection	6.80	3.519917e+08	1.0
4	2555	Bank Central Asia (BCA) Indonesia	Nasabah B	30-Apr-23	NaN	1136.30	3.519920e+08	Interest	1136.30	3.519920e+08	1.0

Gambar 13. Hasil pembacaan dataset model *DecisionTree*

```
# Membersihkan dan memformat data
numeric_columns = ['Original Amount', 'Original Balance', 'Converted Amount', 'Converted Balance', 'Conversion rate']
for col in numeric_columns:
    data[col] = data[col].str.replace('.', '', regex=False) # Hapus pemisah ribuan
    data[col] = data[col].str.replace(',', '.', regex=False) # Ubah koma desimal menjadi titik
    data[col] = pd.to_numeric(data[col], errors='coerce') # Konversi ke numerik

data.info()
data.head()
```

Gambar 14. Pra-proses model *DecisionTree*

```
# Feature engineering untuk kolom 'Date'
data['Date'] = pd.to_datetime(data['Date'], format='%d-%b-%y', errors='coerce')
data['Year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
data['Day'] = data['Date'].dt.day

data.head()
```

Gambar 15. Penyesuaian kolom waktu untuk model *Decision Tree*

Pada Gambar 15, langkah yang dilakukan adalah performatan kolom *date* menjadi tipe data standar, kemudian membuat kolom baru yang digunakan untuk menampung data *date* menjadi format terpisah yaitu *year*, *month*, *day/date*.

```
# Memilih fitur dan target
features = ['Original Balance', 'Converted Balance', 'Year', 'Month', 'Day']
X = data[features]
y = data['Original Amount']

# Menghapus baris dengan nilai NaN
X = X.dropna()
y = y.loc[X.index]
# menghapus kolom dengan nilai NaN
data = data.drop(columns=['Cheque No.'])

data.head()
```

Gambar 16. Pemilihan Fitur untuk Model *DecisionTree*

Langkah selanjutnya yang penulis lakukan seperti pada Gambar 16 yaitu menghapus Baris dan Kolom dengan Nilai NAN. Kemudian dipilih fitur dan target yang akan digunakan untuk pemodelan dengan *DecisionTree*. Sama seperti model KNN, Fitur (*x*) adalah kolom-kolom yang menjadi input untuk prediksi dan Target (*y*) adalah kolom yang ingin diprediksi (pengeluaran nasabah).

```
# Membagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 17. Data latihan dan data uji model *Decision Tree*

Sama seperti pada model KNN, dataset pada model *Decision Tree* dibagi menjadi data latihan dan uji dengan rasio

<https://journal.paramadina.ac.id/index.php/jitc>

Artikel ini adalah artikel dengan akses terbuka, dilisensikan di bawah CC BY 4.0.



tertentu, dengan 20% untuk data pengujian. Data latihan digunakan untuk melatih model, dan data uji untuk mengevaluasi performa model. Syntax `random_state=42` berfungsi sebagai penjamin pembagian data yang sama setiap kali kode dijalankan

```
# Membuat model Decision Tree
model = DecisionTreeRegressor(random_state=42)
model.fit(X_train, y_train)

DecisionTreeRegressor
```

Gambar 18. Penerapan model *Decision Tree*

Pada Gambar 18, diterapkan model *Decision Tree* dengan library *DecisionTreeRegressor*. Library tersebut merupakan algoritma pohon keputusan yang digunakan untuk memprediksi nilai kontinu dalam permasalahan regresi. Berbeda dengan pohon keputusan pada klasifikasi yang menghasilkan label kelas, model ini membagi data ke dalam beberapa cabang berdasarkan nilai fitur, lalu memprediksi nilai output dengan menghitung rata-rata dari nilai target di setiap node akhir (daun). Pendekatan ini memungkinkan model menangkap hubungan non-linear antar variabel, serta memberikan interpretasi yang relatif mudah melalui visualisasi struktur pohon. Dalam konteks penelitian ini, *DecisionTreeRegressor* digunakan untuk memprediksi besarnya pengeluaran nasabah berdasarkan pola data historis yang tersedia.

Setelah model *DecisionTreeRegressor* dilatih menggunakan data latihan, sama seperti pada model KNN, langkah selanjutnya adalah menerapkan model tersebut untuk melakukan prediksi terhadap data uji. Proses ini bertujuan untuk mengukur seberapa baik model mampu memproyeksikan nilai pengeluaran nasabah berdasarkan pola yang telah dipelajari selama pelatihan. Data uji yang digunakan merupakan data yang belum pernah dilihat oleh model sebelumnya, sehingga hasil prediksi mencerminkan kemampuan generalisasi model dalam menghadapi data baru. Nilai-nilai prediksi yang dihasilkan kemudian dibandingkan dengan nilai aktual untuk dievaluasi menggunakan metrik seperti MAE, MSE, dan RMSE seperti pada Gambar 19.

```
# Evaluasi model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)

Mean Absolute Error (MAE): 25052514.445543807
Mean Squared Error (MSE): 7403220299476339.0
Root Mean Squared Error (RMSE): 86041968.24501598
```

Gambar 19. Evaluasi model *Decision Tree*

Sama seperti pada model KNN, pada Gambar 19, penulis melakukan evaluasi model *Decision Tree* menggunakan metrik *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE) dan *Root Mean Squared Error* (RMSE)

```
new_data = [[5000000, 205000000, 5200000, 24000000, 1.03]] # Contoh data baru
prediction = model.predict(new_data)
print("Prediksi pengeluaran:", prediction)

Prediksi pengeluaran: [95046000.]
```

Gambar 20. Uji data baru dengan model *Decision Tree*

Setelah model *DecisionTreeRegressor* berhasil dilatih dan dievaluasi menggunakan data uji, langkah selanjutnya adalah menguji model tersebut dengan data baru yang diinput secara manual. Data ini merepresentasikan kondisi nasabah atau transaksi baru yang belum pernah muncul dalam proses pelatihan maupun pengujian sebelumnya. Dengan memasukkan nilai-nilai fitur secara manual, model akan memproses input tersebut berdasarkan struktur pohon keputusan yang telah terbentuk, kemudian menghasilkan prediksi nilai pengeluaran. Pengujian ini bertujuan untuk melihat seberapa baik model dapat memberikan prediksi pada data yang benar-benar baru, serta menguji akurasi dan konsistensi model dalam penerapan dunia nyata. Sama seperti pada model KNN, terakhir, penulis juga melakukan visualisasi model *Decision Tree* dengan fungsi `plot_tree` seperti pada Gambar 21.

```
# Visualisasi Tree (opsional)
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 8))
plot_tree(model, feature_names=features, filled=True)
plt.savefig("Decesion_tree_regression_actual_vs_predicted.png")
plt.show()
```

Gambar 21. Visualisasi model *Decision Tree*

IV. KESIMPULAN

Penelitian ini berhasil membandingkan kinerja dua model pembelajaran mesin, yaitu *K-Nearest Neighbors* (KNN) dan *Decision Tree*, dalam memprediksi pengeluaran nasabah. Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa Model KNN menunjukkan kinerja yang lebih baik dibandingkan dengan *Decision Tree* dalam hal akurasi dan presisi dalam memprediksi pengeluaran nasabah. Hal ini menunjukkan bahwa KNN lebih efisien dalam menangani data yang lebih kompleks dan memiliki distribusi yang lebih beragam. Namun, *Decision Tree* lebih cepat dalam proses pelatihan dan lebih mudah untuk dipahami, meskipun akurasi sedikit lebih rendah.

Hasil penelitian ini penting bagi praktisi dan peneliti yang ingin mengimplementasikan model pembelajaran mesin untuk analisis perilaku nasabah. Temuan ini memberikan wawasan tentang bagaimana memilih model yang tepat berdasarkan kriteria performa dan kebutuhan komputasi, serta dapat digunakan untuk pengembangan sistem prediksi pengeluaran di sektor keuangan.

Penelitian ini terbatas pada penggunaan dua model saja, sehingga penelitian lebih lanjut dapat mempertimbangkan model pembelajaran mesin lain, seperti *Random Forest* atau *Support Vector Machine*, untuk perbandingan kinerja yang lebih komprehensif. Selain itu, penelitian lanjutan dapat menyelidiki pengaruh fitur tambahan atau data yang lebih besar untuk meningkatkan akurasi model.

REFERENSI

- [1] aws, "Apa itu Machine Learning?," aws.com. Accessed: Jan. 11, 2025. [Online]. Available: <https://aws.amazon.com/id/what-is/machine-learning/>
- [2] S. Amir and Kusriani, "Prediksi Kebutuhan Nasabah Dengan Teknik Data Mining Dalam Mendukung Strategi Pemasaran Bank," *Jurnal Bangkit Indonesia*, vol. 7, no. 2, p. 40, 2018.
- [3] N. Nuraeni, "Klasifikasi Data Mining untuk Prediksi Potensi Nasabah dalam Membuat Deposito Berjangka Data Mining Classification for Predicting Customer Potential in Making Term Deposits," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 3, no. 01, pp. 65–75, 2021.
- [4] N. M. Wasilah, "Penerapan Data Mining Untuk Menganalisis Kepuasan Konsumen Menggunakan Metode C4.5," vol. 5, no. 1, pp. 85–90, 2013.
- [5] R. Indah, O. Limabri, and A. Arif, "Reliya Indah Okta Limabri, Ferry Putrawansyah, Alfis Arif PENERAPAN DATA MINING UNTUK MENGLASIFIKASI NASABAH BANK SUMSEL BABEL MENGGUNAKAN ALGORITMA C4.5," *ESCAF 2nd2023*, pp. 1101–1108, 2023.
- [6] F. Febri, Hasbi Yallah, Muhammad Darwis, and Retno Henrowati, "Penerapan Algoritma Id3 Melalui Aplikasi Orange Untuk Prediksi Akurasi Akreditasi Sekolah Dasar Di Depok," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 7, no. 2, pp. 131–141, 2024, doi: 10.36595/misi.v7i2.1199.
- [7] A. Suresh, "Prediction of major crop yields of Tamilnadu using K-means and Modified KNN," *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, no. Icces, pp. 88–93, 2018.
- [8] M. Yusa, E. Utami, and E. Luthfi. Taufiq, "Evaluasi Performa Algoritma Klasifikasi Decision Tree Id3," *InfoSys Journal*, vol. 4, no. 1, pp. 23–34, 2016.
- [9] A. Chakure, "Decision Tree Classification," *Toward Data Science*. [Online]. Available: <https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac>
- [10] A. A. Suryanto, "Penerapan Metode Mean Absolute Error (Mea) Dalam Algoritma Regresi Linear Untuk Prediksi Produksi Padi," *Saintekbu*,

- vol. 11, no. 1, pp. 78–83, 2019, doi:
10.32764/saintekbu.v11i1.298.
- [11] Widiarti, R. R. Periwati, and A. Sutrisno, “Perbandingan Mean Squared Error (MSE) Metode Prasad-Rao dan Jiang-Lahiri-Wan Pada Pendugaan Area Kecil,” *Seminar Nasional Teknoka*, vol. 2, no. 2502, pp. 56–60, 2017, [Online]. Available: <https://journal.uhamka.ac.id/index.php/teknoka/article/view/752/296>
- [12] E. K. Hendarwati, P. Lepong, and S. Suyitno, “Pemilihan Semivariogram Terbaik Berdasarkan Root Mean Square Error (RMSE) pada Data Spasial Eksplorasi Emas Awak Mas,” *Geosains Kutai Basin*, vol. 6, no. 1, p. 47, 2023, doi: 10.30872/geofisunmul.v6i1.1072.
- [13] dicoding intern, “Apa itu Machine Learning? Beserta Pengertian dan Cara Kerjanya,” [dicoding.com](https://www.dicoding.com). Accessed: Jan. 11, 2025. [Online]. Available: <https://www.dicoding.com/blog/machine-learning-adalah/>
- [14] IBM, “Apa algoritma k-tetangga terdekat (KNN)?,” [ibm.com](https://www.ibm.com). Accessed: Jan. 11, 2025. [Online]. Available: <https://www.ibm.com/id-id/topics/knn>
- [15] Siti Khadijah Azzukhruf Firdausi, “Decision Tree: Pengertian, Manfaat, Plus-Minus, & Contohnya,” dibimbing.id. Accessed: Jan. 11, 2025. [Online]. Available: <https://dibimbing.id/blog/detail/decision-tree-pengertian-manfaat-plus-minus-contohnya>