LAPORAN TUGAS BESAR PRAKTIKUM
MACHINE LEARNING
KLASIFIKASI DETEKESI TRANSAKSI MENCURIGAKAN

Disusun Oleh:

Ariel Adriazul Ahwan	F55123071
Asfita Saldarisya Nadjun	F55123072
Muh. Alfaiz	F55123085
Fahri Alamsyah	F55123099

Kelas TI C

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2025

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sektor keuangan, khususnya perbankan dan penyedia layanan pembayaran, merupakan salah satu industri yang paling rentan terhadap aktivitas ilegal seperti penipuan (fraud). Seiring dengan peningkatan volume transaksi digital dan adopsi teknologi finansial, modus operandi penipuan menjadi semakin canggih dan sulit dideteksi secara manual. Dampak dari penipuan transaksi tidak hanya merugikan secara finansial bagi individu dan institusi, tetapi juga dapat merusak reputasi, menurunkan kepercayaan pelanggan, dan bahkan berimplikasi pada stabilitas ekonomi makro. Berbagai laporan menunjukkan bahwa kerugian akibat penipuan transaksi global mencapai miliaran dolar setiap tahunnya, dan angka ini terus meningkat.

Secara tradisional, deteksi transaksi mencurigakan seringkali dilakukan menggunakan aturan berbasis heuristik (rule-based systems) atau tinjauan manual oleh analis. Meskipun metode ini memiliki keunggulan dalam identifikasi pola yang jelas, namun terbatas dalam menangani volume data yang besar, pola penipuan yang kompleks, dan sifat penipuan yang adaptif. Sistem berbasis aturan seringkali menghasilkan tingkat *false positives* (transaksi sah yang diklasifikasikan sebagai penipuan) yang tinggi, yang menyebabkan pemblokiran transaksi yang tidak perlu dan mengganggu pengalaman pengguna. Di sisi lain, *false negatives* (transaksi penipuan yang tidak terdeteksi) dapat mengakibatkan kerugian finansial yang signifikan.

Kemajuan pesat dalam bidang *machine learning* telah membuka peluang baru untuk mengatasi keterbatasan metode deteksi penipuan konvensional. Algoritma *machine learning* memiliki kemampuan untuk mempelajari pola-pola kompleks dari data transaksi historis, termasuk karakteristik transaksi yang sah dan transaksi yang mencurigakan. Dengan memanfaatkan teknik seperti klasifikasi, *clustering*, dan deteksi anomali, model *machine learning* dapat secara otomatis mengidentifikasi perilaku transaksi yang menyimpang dari norma, bahkan untuk pola penipuan yang belum pernah terlihat sebelumnya.

1.2 Rumusan Masalah

Transaksi kartu kredit yang bersifat curang (fraudulent) merupakan masalah serius yang merugikan baik nasabah maupun penyedia layanan keuangan. Dataset transaksi kartu kredit umumnya memiliki ketidakseimbangan kelas yang sangat signifikan, di mana jumlah transaksi normal jauh lebih banyak daripada transaksi curang. Hal ini menjadi tantangan utama dalam membangun model klasifikasi yang efektif, karena model cenderung lebih baik dalam memprediksi kelas mayoritas (transaksi normal) dan kurang sensitif terhadap kelas minoritas (transaksi curang) yang justru lebih penting untuk dideteksi.

Berdasarkan hal tersebut, maka rumusan masalahnya adalah:

1. Bagaimana karakteristik distribusi data transaksi kartu kredit, khususnya terkait proporsi antara transaksi normal dan transaksi curang, serta bagaimana distribusi fitur-fitur penting seperti 'Amount' dan 'Time'?
2. Bagaimana teknik *oversampling* seperti SMOTE (Synthetic Minority Over-sampling Technique) dapat membantu mengatasi masalah ketidakseimbangan kelas dalam dataset transaksi kartu kredit untuk meningkatkan kinerja pendeteksian transaksi curang?
3. Bagaimana performa model *Random Forest Classifier* dalam mengklasifikasikan transaksi kartu kredit sebagai normal atau curang setelah dilakukan normalisasi fitur dan penanganan ketidakseimbangan kelas menggunakan SMOTE, diukur dengan metrik evaluasi seperti *confusion matrix, classification report*, dan *ROC-AUC score*?

1.3 Tujuan

Berdasarkan rumusan masalah di atas, tujuan dari analisis dan pemodelan ini adalah:

1. Melakukan analisis eksploratif data (EDA) untuk memahami distribusi kelas target (transaksi normal vs. curang), statistik deskriptif fitur 'Amount', distribusi fitur 'Time', serta korelasi antar fitur dengan kelas target.
2. Menerapkan normalisasi fitur menggunakan *StandardScaler* untuk menyeragamkan skala data pada fitur-fitur numerik.

3. Mengimplementasikan teknik *oversampling* SMOTE pada data latih untuk menyeimbangkan distribusi kelas antara transaksi normal dan transaksi curang.
4. Membangun model klasifikasi menggunakan algoritma *Random Forest Classifier* untuk memprediksi apakah suatu transaksi kartu kredit bersifat curang atau normal. Mengevaluasi kinerja model *Random Forest Classifier* yang telah dilatih menggunakan metrik *confusion matrix, classification report*, dan *ROC-AUC score* untuk menilai kemampuannya dalam mendeteksi transaksi mencurigakan secara akurat, terutama dalam mengidentifikasi transaksi curang (kelas minoritas).

BAB II
PEMBAHASAN

2.1 Model

2.1.1 Klasifikasi

Klasifikasi digunakan untuk memisahkan atau mengelompokkan data transaksi menjadi dua kategori, yaitu transaksi normal dan transaksi fraud (penipuan). Proses ini dilakukan dengan membangun model machine learning yang mempelajari pola-pola dari data historis transaksi. Pola ini diperoleh dari berbagai fitur numerik yang telah ditransformasi sebelumnya, termasuk informasi jumlah uang dalam transaksi (Amount), waktu transaksi (Time), dan fitur-fitur hasil ekstraksi seperti V1 hingga V28.

Tujuan dari penerapan klasifikasi dalam konteks ini adalah agar model dapat secara otomatis mengenali karakteristik transaksi yang berisiko atau mencurigakan. Hal ini sangat penting dalam dunia keuangan untuk membantu mencegah kerugian akibat penipuan. Klasifikasi dipilih karena jenis data target yang digunakan bersifat kategorikal biner, yaitu 0 untuk transaksi normal dan 1 untuk transaksi fraud. Dengan kata lain, model akan dilatih untuk memberikan prediksi apakah suatu transaksi tergolong aman atau mencurigakan berdasarkan fitur-fitur yang dimilikinya.

Metode klasifikasi seperti Random Forest digunakan karena mampu mengelola kompleksitas fitur dalam data, menghasilkan prediksi yang akurat, serta memiliki keunggulan dalam menangani data tidak seimbang seperti kasus fraud detection, di mana jumlah transaksi fraud jauh lebih sedikit dibanding transaksi normal

1. Diskusi tentang alasan pemilihan model yang digunakan.

   Model yang digunakan dalam proyek ini adalah Random Forest Classifier. Model ini dipilih karena sangat cocok untuk tugas klasifikasi biner seperti deteksi transaksi penipuan, terutama saat data bersifat kompleks dan tidak linear. Random Forest juga terkenal karenakemampuannya untuk mengatasi overfitting dan bekerja dengan baik pada data yang tidak seimbang.

2. Pertimbangan seperti kompleksitas data, tujuan analisis, dan interpretabilitas mode.
    a. Kompleksitas Data: Dataset memiliki 30 fitur numerik yang telah ditransformasi melalui PCA. Struktur datanya tidak linear dan sulit untuk dianalisis secara sederhana. Random Forest cocok untuk menangani interaksi antar fitur semacam ini.
    b. Tujuan Analisis: Tujuan utama adalah mendeteksi transaksi penipuan secara akurat, meskipun jumlah kasus fraud sangat sedikit (imbalanced class). Random Forest mendukung pemodelan probabilistik dan dapat menangani class imbalance dengan baik menggunakan parameter seperti class_weight='balanced'
    c. Interpretabilitas Model: Meskipun tidak seinterpretable Logistic Regression, Random Forest tetap memberikan feature importance sehingga kita masih bisa mengetahui fitur mana yang paling berpengaruh terhadap keputusan model.
3. Penjelasan tentang kelebihan dan kekurangan model yang dipilih.

    Kelebihan :
    a. Kuat terhadap overfitting karena menggunakan banyak pohon keputusan
    b. Bekerja baik dengan data numerik maupun kategorikal
    c. Dapat menangani data yang tidak terdistribusi normal.
    d. Mendukung pengukuran feature importance untuk interpretasi hasil.

    Kekurangan :
    a. Kurang efisien dalam komputasi dibanding model sederhana (misalnya Logistic Regression).
    b. Interpretasi hasil lebih sulit dibanding model linier.
    c. Jika tidak diatur dengan baik, waktu pelatihan dan inferensi bisa lama pada dataset besar.

2.2 Dataset

2.2.1 Kualitas dan Relevansi Dataset yang Digunakan

Dataset yang digunakan merupakan dataset transaksi kartu kredit yang banyak digunakan dalam penelitian deteksi penipuan. Dataset ini terdiri dari transaksi yang dilakukan oleh nasabah kartu kredit Eropa selama dua hari pada bulan September 2013

1. Dataset memiliki fitur-fitur hasil transformasi PCA (V1-V28), yang berarti data sudah melalui proses feature extraction awal untuk menjaga kerahasiaan pengguna.
2. Terdapat dua fitur utama yang tidak ditransformasi:
    a. Amount: jumlah transaksi.
    b. Time: waktu transaksi dalam detik sejak transaksi pertama.
3. Target Class menunjukkan label:
    a. 0 = transaksi normal
    b. 1 = transaksi fraud

2.2.2 Penggunaan dataset yang cukup besar untuk analisis yang efektif

Dataset terdiri dari 284.807 baris data, dengan:

1. Sekitar 492 transaksi penipuan (fraud).
2. Sisanya adalah transaksi normal.

Ukuran dataset ini cukup besar dan realistis untuk membangun model klasifikasi berbasis machine learning, sekaligus menantang karena data tidak seimbang (imbalanced), yang merupakan tantangan umum dalam deteksi penipuan

2.2.3 Penjelasan tentang sumber dataset dan alasan pemilihan

Dataset yang digunakan dalam proyek ini berasal dari situs Kaggle, tepatnya dari repositori berjudul *"Credit Card Fraud Detection"* yang disediakan oleh Machine Learning Group dari Université Libre de Bruxelles (ULB). Dataset ini merupakan data transaksi kartu kredit yang dikumpulkan dari pelanggan Eropa selama dua hari pada bulan September 2013. Link dataset: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Alasan Pemilihan Dataset:

1. Relevan dengan topik klasifikasi: Dataset ini dirancang untuk mendeteksi transaksi penipuan (fraud), yang merupakan kasus klasifikasi biner — sesuai dengan tujuan utama proyek.
2. Data nyata dan realistis: Data ini berasal dari transaksi asli di dunia nyata, sehingga dapat mencerminkan tantangan sebenarnya dalam mendeteksi anomali dalam data keuangan.
3. Tantangan data tidak seimbang: Karena jumlah transaksi penipuan sangat kecil dibandingkan dengan transaksi normal, dataset ini memberikan kesempatan untuk menerapkan teknik penanganan data imbalanced, seperti SMOTE, class weighting, dan analisis metrik beyond accuracy (precision, recall, F1-score).
4. Kualitas data terjaga: Semua fitur (V1–V28) merupakan hasil transformasi Principal Component Analysis (PCA) untuk menjaga privasi data pengguna, sehingga sudah bebas dari informasi sensitif namun tetap berguna secara statistik.
5. Dukungan komunitas dan dokumentasi: Karena dataset ini populer, tersedia banyak referensi, baseline model, dan dokumentasi yang dapat digunakan sebagai acuan dan pembanding dalam membangun dan mengevaluasi model.

2.3 Preprocessing & Feature Extraction

Dalam proyek deteksi transaksi penipuan ini, tahap *preprocessing* dan *feature extraction* menjadi bagian penting untuk memastikan data yang digunakan dalam pelatihan model bersih, terstruktur, dan relevan

1. Penerapan teknik preprocessing yang tepat, termasuk:
    a. Penanganan missing values

        Proses preprocessing dimulai dengan penanganan missing values, yang dilakukan dengan memeriksa apakah terdapat nilai kosong atau NaN dalam dataset. Berdasarkan hasil eksplorasi data, tidak ditemukan missing values dalam dataset, sehingga tidak diperlukan imputasi atau pembuangan data.
    b. Normalisasi atau standarisasi data

Langkah selanjutnya adalah standarisasi data, khususnya pada fitur-fitur numerik seperti Amount dan Time, yang tidak termasuk dalam hasil transformasi PCA. Standarisasi dilakukan menggunakan StandardScaler untuk memastikan bahwa setiap fitur memiliki skala distribusi yang seragam, yaitu rata-rata 0 dan standar deviasi 1. Hal ini penting karena algoritma seperti Random Forest dapat dipengaruhi oleh perbedaan skala fitur, terutama pada dataset yang sangat tidak seimbang.

c. Pengkodean variabel kategorikal, dll

Dataset ini hanya terdiri dari fitur numerik (karena sudah melalui proses PCA), sehingga tidak ada fitur kategorikal yang perlu dikodekan. Maka, teknik seperti one-hot encoding tidak diperlukan dalam kasus ini.

2. Penjelasan tentang setiap langkah preprocessing yang dilakukan

a. Pemeriksaan missing values

Dataset dibersihkan dari nilai kosong secara otomatis atau tidak ditemukan nilai NaN.

b. Standarisasi fitur Amount dan Time

Dilakukan menggunakan StandardScaler agar fitur tersebut tidak mendominasi model karena skala yang berbeda.

c. Pemisahan data fitur dan label

Variabel fitur (X) dan target (y) dipisahkan, lalu dilakukan pembagian data menjadi data latih dan uji.

d. Stratified Train-Test Split

Untuk menjaga proporsi kelas fraud dan normal tetap seimbang antara data latih dan data uji, digunakan parameter stratify=y.

3. Penerapan teknik feature extraction yang relevan

Pada dataset ini, teknik feature extraction sudah diterapkan sebelumnya oleh penyedia data. Fitur-fitur V1 hingga V28 merupakan hasil dari proses Principal Component Analysis (PCA) pada fitur asli (karena alasan privasi

dan keamanan data pengguna kartu kredit). Sehingga tidak perlu dilakukan feature extraction tambahan.

4. Penjelasan tentang fitur yang diambil dan alasan pemilihan fitur tersebut
   a. Fitur V1–V28: hasil transformasi PCA dari data asli. Dipertahankan karena telah mewakili dimensi penting dari data transaksi.
   b. Fitur Amount: tetap digunakan karena berkaitan langsung dengan jumlah transaksi, yang bisa menjadi indikator utama penipuan.
   c. Fitur Time: digunakan karena bisa menunjukkan pola penipuan berdasarkan waktu tertentu (misal, transaksi mencurigakan di tengah malam).

   Semua fitur ini digunakan karena dianggap memiliki kontribusi dalam membedakan transaksi normal dan penipuan.
5. Penggunaan teknik seperti PCA (Principal Component Analysis) jika relevan

   Teknik PCA sudah diterapkan oleh penyedia dataset sebelum data dirilis ke publik. Hal ini dilakukan untuk:
   a. Menjaga privasi data asli pengguna.
   b. Mengurangi dimensi data dan menghilangkan redundansi.
   c. Meningkatkan efisiensi pelatihan model.

   Karena PCA telah dilakukan sebelumnya dan fitur-fitur hasilnya sudah tersedia, tidak perlu dilakukan PCA ulang dalam proses ini.

2.4 Exploratory Data Analysis (EDA)

1. Penyajian visualisasi data yang informatif.

   Visualisasi digunakan secara efektif untuk membantu memahami distribusi dan karakteristik data:
   a. Countplot kelas (normal vs fraud)

      Menampilkan perbandingan jumlah transaksi normal dan penipuan. Terlihat ketimpangan data yang sangat besar (transaksi normal jauh lebih banyak), yang akan mengindikasikan masalah pada imbalanced class.
   b. Histogram fitur Amount

Memvisualisasikan distribusi jumlah transaksi. Sebagian besar transaksi bernilai kecil, namun terdapat beberapa transaksi dengan jumlah yang sangat besar (outlier).

c. Histogram fitur Time

Menunjukkan distribusi transaksi berdasarkan waktu. Dapat digunakan untuk mengamati apakah ada pola waktu tertentu di mana transaksi penipuan lebih sering terjadi.

d. Heatmap korelasi dengan target (Class)

Menampilkan korelasi antara setiap fitur dengan kelas (fraud atau tidak). Memberikan gambaran fitur mana yang paling berpengaruh terhadap prediksi penipuan.

e. Confusion Matrix

Menampilkan performa model klasifikasi secara visual, memperlihatkan jumlah prediksi benar dan salah untuk masing-masing kelas.

f. ROC Curve

Memberikan gambaran performa model berdasarkan keseimbangan antara true positive rate dan false positive rate.

2. Analisis statistik deskriptif untuk memahami distribusi data.

Statistik deskriptif dilakukan khususnya pada fitur Amount:

a. Ditampilkan nilai mean, standard deviation, min, max, dan quartiles.
b. Analisis ini menunjukkan bahwa sebagian besar transaksi memiliki jumlah kecil, namun ada nilai maksimum yang sangat besar, mengindikasikan kehadiran outlier.

3. Identifikasi pola, tren, dan outlier dalam dataset.

a. Dari visualisasi dan statistik fitur Amount, dapat terlihat bahwa sebagian kecil transaksi memiliki nilai sangat besar, yang bisa menjadi indikasi penipuan atau outlier.
b. Korelasi antara fitur-fitur hasil PCA (V1–V28) dengan variabel Class memberikan informasi tentang pola hubungan linier antara dimensi tersembunyi data dan kejadian fraud.

c. Distribusi waktu (Time) memungkinkan analisis tren temporal, seperti peningkatan jumlah fraud pada waktu-waktu tertentu dalam sehari.

BAB III
PROSEDUR KERJA

3.1 Langkah Kerja

1. *Import library*

```
# Import library
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
```

Bagian ini memuat semua pustaka Python yang dibutuhkan:

a. pandas: untuk manipulasi data.

b. train_test_split: untuk membagi data ke data latih dan uji.

c. StandardScaler: untuk menstandarisasi fitur numerik.

d. SMOTE: teknik oversampling untuk menangani data tidak seimbang.

e. RandomForestClassifier: algoritma model klasifikasi.

f. classification_report, confusion_matrix, roc_auc_score: metrik evaluasi model.

2. Load persiapan data

```
# Pastikan dataset bersih dari NaN di kolom target
df = pd.read_csv('creditcard.csv')
df = df.dropna(subset=['Class'])
```

a. Dataset dibaca dari file creditcard.csv.

b. Data yang memiliki nilai kosong (NaN) pada kolom Class dihapus.

3. Pisahkan Fitur dan Target

```
# Pisahkan fitur dan target
X = df.drop('Class', axis=1)
y = df['Class']
```

a. X: semua fitur (independen) selain kolom Class.

b. y: target (kolom Class, 0 = normal, 1 = fraud).

4. Split Data dan Normalisasi

```
# Lanjutkan train_test_split seperti sebelumnya
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)

# 4. Normalisasi fitur (kecuali 'Time' jika mau dikecualikan)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

a. Data dibagi 80% untuk training, 20% testing. Dataset dibagi 80% untuk pelatihan dan 20% untuk pengujian. stratify=y menjaga proporsi fraud dan normal tetap sama di train/test.

b. Standarisasi dilakukan agar semua fitur berada dalam skala yang sama, penting untuk beberapa algoritma ML.StandardScaler digunakan untuk menstandarisasi nilai fitur. Ini penting karena banyak algoritma ML sensitif terhadap skala fitur.

5. Penanganan Data Tidak Seimbang

```
# 5. Tangani imbalance dengan SMOTE (oversampling)
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train_scaled, y_train)
```

Karena data fraud sangat sedikit, digunakan SMOTE (Synthetic Minority SMOTE membuat data sintetis dari kelas minoritas (fraud) dengan membentuk titik-titik baru di sekitar sampel aslinya. Oversampling Technique) untuk menyeimbangkan jumlah kelas Hasil: kelas 1 (fraud) kini seimbang jumlahnya dengan kelas 0.

6. Training Model

```
# 6. Training model Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_res, y_train_res)
```

Model Random Forest dilatih menggunakan data yang sudah diseimbangkan. Model RandomForestClassifier dengan 100 pohon decision tree dilatih pada data yang sudah diseimbangkan oleh SMOTE. Random Forest dipilih karena kuat terhadap overfitting, bekerja baik untuk data dengan banyak fitur, dan bisa mengukur pentingnya fitur.

7. Evaluasi Model

```
# 7. Prediksi dan evaluasi
y_pred = model.predict(X_test_scaled)
y_prob = model.predict_proba(X_test_scaled)[:, 1]

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print(f"ROC-AUC Score: {roc_auc_score(y_test, y_prob):.4f}")
```

   a. Prediksi dilakukan pada data uji.
   b. Evaluasi mencakup:
   c. Confusion Matrix
   d. Classification Report (akurasi, precision, recall, F1)
   e. ROC-AUC Score (kemampuan membedakan fraud dan normal)

8. Visualisasi Distribusi Kelas

```
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Cek proporsi kelas fraud dan normal
print("Distribusi kelas:")
print(y.value_counts(normalize=True))
plt.figure(figsize=(6,4))
sns.countplot(x=y)
plt.title("Distribusi Kelas: Normal vs Fraud")
plt.show()
```

   Menampilkan grafik jumlah data fraud dan normal. Menampilkan proporsi kelas (0 dan 1) untuk melihat imbalance. Grafik batang (countplot) menunjukkan jumlah transaksi normal vs fraud.

9. Statistik Deskriptif dan Visualisasi Fitur

```
# 2. Statistik deskriptif fitur 'Amount' (jumlah transaksi)
print("\nStatistik fitur Amount:")
print(df['Amount'].describe())

plt.figure(figsize=(8,4))
sns.histplot(df['Amount'], bins=50, kde=True)
plt.title("Distribusi Jumlah Transaksi (Amount)")
plt.show()
```

   Menampilkan statistik dan distribusi jumlah transaksi (Amount) dan waktu (Time). Statistik deskriptif fitur Amount seperti mean, min, max, std.

Histogram untuk Amount dan Time membantu memahami distribusi transaksi.

10. Visualisasi fitur time

```
# 3. Visualisasi fitur 'Time' (waktu transaksi dalam detik sejak awal rekaman)
plt.figure(figsize=(8,4))
sns.histplot(df['Time'], bins=50, kde=True)
plt.title("Distribusi Waktu Transaksi (Time)")
plt.show()
```

memvisualisasikan distribusi waktu transaksi (Time) dalam dataset. Fitur Time menunjukkan jumlah detik sejak transaksi pertama dicatat

a. plt.figure(figsize=(8,4)) Mengatur ukuran figure (grafik) agar lebar 8 inci dan tinggi 4 inci, membuat tampilan lebih proporsional dan mudah dibaca.
b. sns.histplot(df['Time'], bins=50, kde=True)Membuat histogram dari kolom Time menggunakan 50 *bin* (interval). kde=True menambahkan kurva KDE (Kernel Density Estimation) di atas histogram untuk menunjukkan distribusi data secara halus. Tujuannya untuk melihat kapan transaksi lebih banyak terjadi—misalnya apakah terkonsentrasi di awal, tengah, atau akhir periode rekaman.

11. Korelasi Fitur

```
# 4. Heatmap korelasi fitur dengan target 'Class'
plt.figure(figsize=(12,8))
corr = df.corr()
sns.heatmap(corr[['Class']].sort_values(by='Class', ascending=False), annot=True, cmap='coolwarm')
plt.title("Korelasi Fitur dengan Kelas Fraud")
plt.show()
```

Menggunakan heatmap untuk melihat fitur mana yang paling berkorelasi dengan Class (fraud). Menghitung korelasi antar fitur dengan target

(Class). Heatmap menampilkan fitur yang paling berkorelasi (positif/negatif) dengan fraud.

12. Visualisasi Confusion Matrix

```
#·5.·Visualisasi·Confusion·Matrix·hasil·model
from·sklearn.metrics·import·ConfusionMatrixDisplay

ConfusionMatrixDisplay.from_estimator(model,·X_test_scaled,·y_test,·cmap='Blues')
plt.title("Confusion·Matrix·Model·Random·Forest")
plt.show()
```

Menampilkan confusion matrix dalam bentuk visual untuk melihat distribusi prediksi benar/salah. Visualisasi confusion matrix agar lebih mudah dipahami, membandingkan prediksi dan kenyataan.

13. Visualisasi ROC Curve

```
# 6. Visualisasi ROC Curve
from sklearn.metrics import RocCurveDisplay

RocCurveDisplay.from_estimator(model, X_test_scaled, y_test)
plt.title("ROC Curve Model Random Forest")
plt.show()
```

Menampilkan kurva ROC untuk melihat performa klasifikasi di berbagai threshold ROC Curve menunjukkan trade-off antara true positive rate dan false positive rate di berbagai threshold klasifikasi. Area di bawah kurva (AUC) memberi skor kinerja model dalam membedakan kelas.

BAB IV

HASIL

4.1 Hasil Output

1. Tampilan Training

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Setelah bagian ini dijalankan, model Random Forest telah selesai **dilatih** dan siap untuk digunakan dalam melakukan prediksi terhadap data uji (X_test_scaled). Tidak ada output langsung yang dicetak ke layar dari kode ini, tetapi model RandomForestClassifier sekarang menyimpan struktur pohon dan dapat memprediksi dengan metode .predict() atau .predict_proba()

2. Tampilan Evaluasi Model

```
Confusion Matrix:
[[20568     4]
 [    7    39]]

Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     20572
         1.0       0.91      0.85      0.88        46

    accuracy                           1.00     20618
   macro avg       0.95      0.92      0.94     20618
weighted avg       1.00      1.00      1.00     20618

ROC-AUC Score: 0.9820
```

Interpretasi:

a. **True Negatives (TN)**: 20.568 transaksi normal diklasifikasikan benar sebagai normal.

b. **False Positives (FP)**: 4 transaksi normal salah diklasifikasikan sebagai fraud.

c. **False Negatives (FN)**: 7 transaksi fraud tidak terdeteksi (salah diklasifikasikan sebagai normal).

d. **True Positives (TP)**: 39 transaksi fraud berhasil terdeteksi.

Model berhasil mendeteksi 39 dari 46 kasus fraud, hanya meleset 7 kasus. Ini menunjukkan kinerja yang sangat baik, mengingat kasus fraud sangat sedikit (klas minoritas).

Penjelasan per kelas:

1. **Kelas 0.0 (Normal):**
    a. **Precision 1.00**: Semua prediksi untuk kelas normal benar.
    b. **Recall 1.00**: Semua transaksi normal berhasil dikenali dengan benar.
    c. **F1-score 1.00**: Sangat tinggi, mencerminkan keseimbangan precision dan recall.
2. **Kelas 1.0 (Fraud):**
    a. **Precision 0.91**: Dari semua transaksi yang diprediksi fraud, 91% benar-benar fraud.
    b. **Recall 0.85**: Dari semua transaksi fraud yang sebenarnya, 85% berhasil dikenali.
    c. **F1-score 0.88**: Mengindikasikan keseimbangan yang baik, meski masih ada 15% fraud yang terlewat.

Akurasi total:

1. 100% (dibulatkan) dari seluruh data (20.618 transaksi) diprediksi dengan sangat baik.

    Macro avg dan Weighted avg juga tinggi, menunjukkan performa stabil meskipun data tidak seimbang.
    a. Nilai ROC-AUC = 0.9820 sangat tinggi.
    b. Menunjukkan bahwa model sangat baik membedakan antara transaksi normal dan fraud.
    c. Semakin mendekati 1, semakin baik model dalam mengenali kelas positif (fraud) meski minoritas.

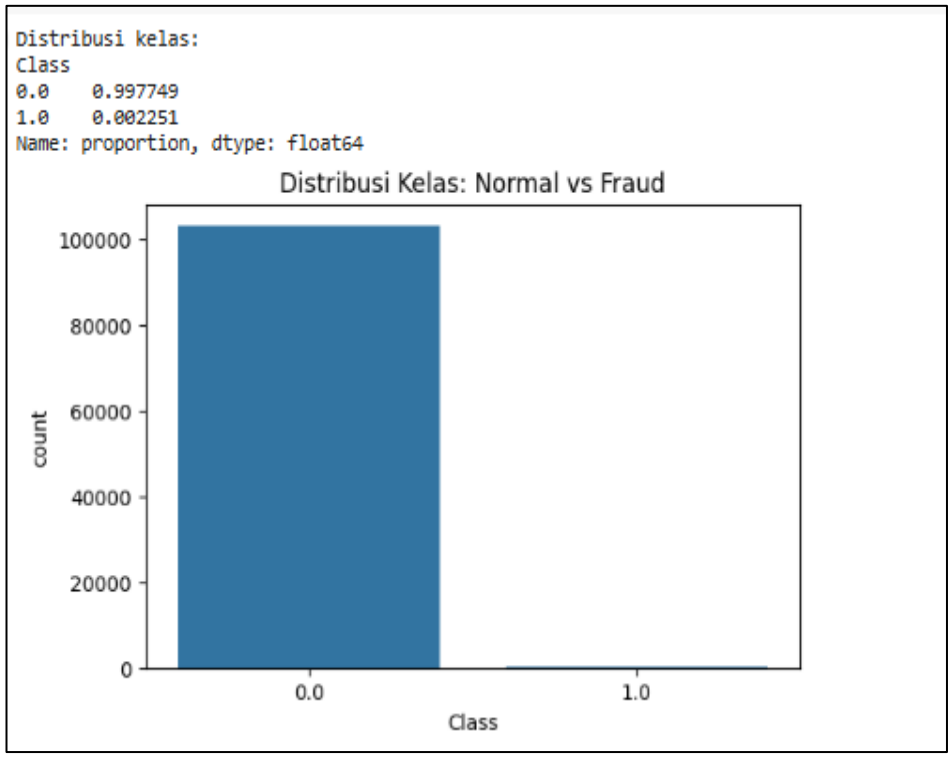
    Model Random Forest yang digunakan:
    a. Sangat akurat untuk mendeteksi transaksi normal.
    b. Efektif dalam mendeteksi fraud, dengan hanya sedikit kesalahan.

c. Presisi dan recall yang tinggi pada kelas fraud menunjukkan bahwa model bisa digunakan secara praktis untuk sistem deteksi penipuan.
d. ROC-AUC yang sangat baik (0.9820) memperkuat kepercayaan bahwa model ini mampu menangani data tidak seimbang dan mengenali pola penipuan secara efektif.

Jika dibutuhkan, model ini bisa lebih ditingkatkan dengan:

a. Analisis outlier lebih lanjut.
b. Tuning parameter Random Forest.
c. Penggabungan lebih banyak fitur/engineering.

3. Tampilan Visualisasi Distribusi kelas



99.77% data adalah transaksi normal (Class = 0) Hanya 0.23% data adalah transaksi fraud (Class = 1) Grafik Batang (Bar Chart):

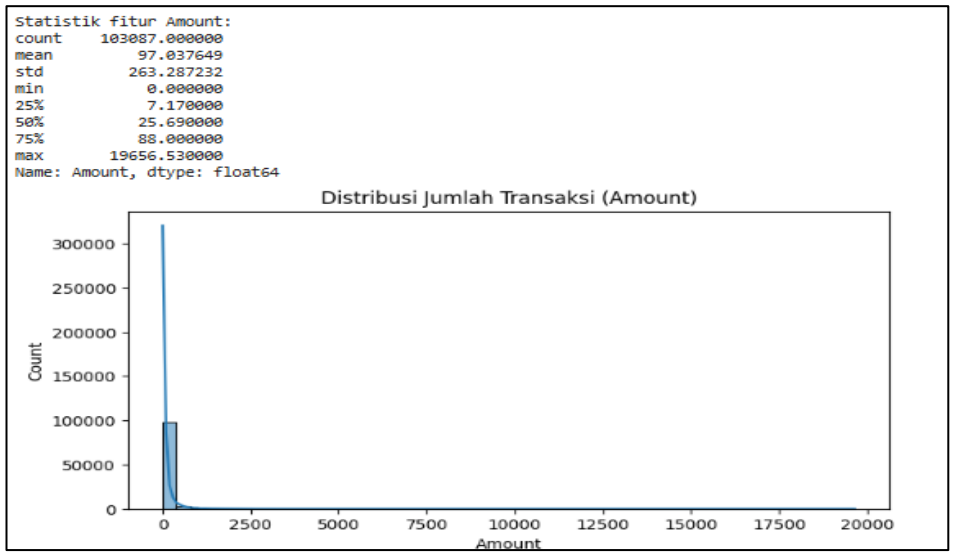
a. Batang kiri mewakili kelas 0 (normal), sangat tinggi, menunjukkan dominasi besar data normal.
b. Batang kanan mewakili kelas 1 (fraud), hampir tidak terlihat karena jumlahnya sangat kecil.

Dataset sangat tidak seimbang (imbalanced) — kasus umum dalam data fraud.

Kondisi ini berbahaya jika tidak ditangani, karena model bisa “malas” dan hanya memprediksi semua sebagai normal.

Oleh karena itu, langkah SMOTE (Synthetic Minority Over-sampling Technique) diperlukan (dan sudah dilakukan dalam kode) untuk menyeimbangkan kelas selama training.

4. Tampilan Statistik Deskriptif dan Visualisasi Fitur



Grafik ini menghasilkan ringkasan statistik deskriptif untuk kolom Amount, meliputi:

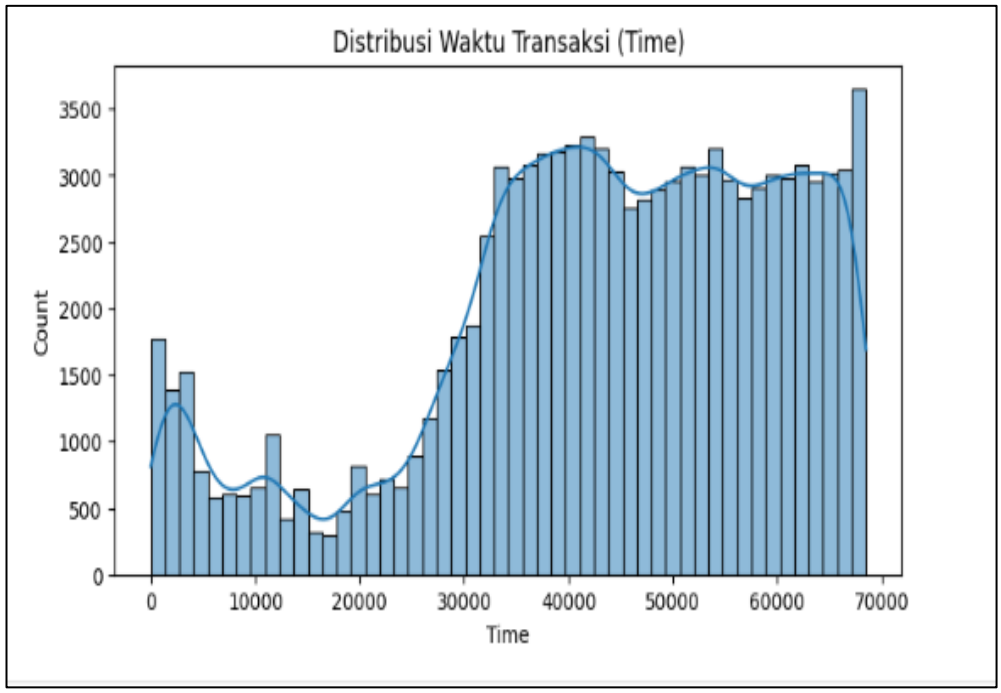
a. **count**: Jumlah data (banyaknya transaksi yang tercatat).
b. **mean**: Rata-rata jumlah transaksi.
c. **std**: Standar deviasi, menunjukkan seberapa tersebar nilai Amount.
d. **min / max**: Nilai transaksi terkecil dan terbesar.
e. **25% / 50% (median) / 75%**: Kuartil pertama, median, dan kuartil ketiga — menunjukkan persebaran data.
f. **bins=50**: Data dikelompokkan dalam 50 kelompok interval nilai.
g. **kde=True**: Ditambahkan kurva KDE (Kernel Density Estimate), yaitu pendekatan kurva halus untuk distribusi.
h. Visualisasi ini membantu kita melihat:

1. Apakah data Amount terdistribusi normal atau miring.={
2. Apakah ada **outlier** (nilai ekstrim) seperti transaksi dengan jumlah sangat besar.

Hasil visualisasi ini menunjukkan

1. Distribusi kemungkinan **tidak normal (right-skewed)** — mayoritas transaksi berjumlah kecil, hanya sedikit yang sangat besar.
2. Bisa terlihat adanya **outlier** seperti transaksi dengan nilai sangat tinggi.
3. Ini penting untuk:
4. Menentukan apakah perlu **transformasi (misal log-scaling)**.
5. Mengetahui apakah Amount memengaruhi status fraud.

5. Tampilan Visualisasi fitur 'Time'



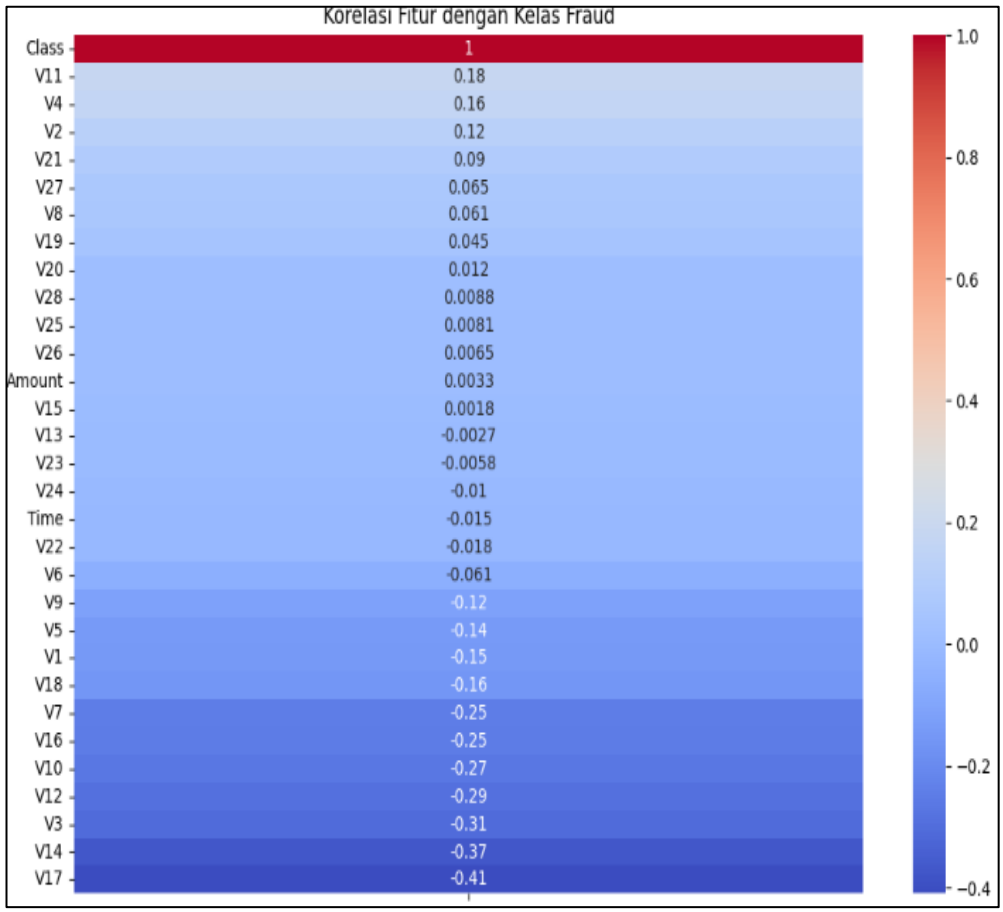
Tujuan visualisasi: Mengetahui kapan transaksi dilakukan lebih sering dalam periode waktu tertentu.

a. Karena Time diukur dalam detik sejak awal perekaman, grafik ini dapat mengungkap:
    a. Pola volume transaksi harian (misalnya: jam sibuk, aktivitas tinggi di pagi atau malam hari).
    b. Apakah fraud cenderung terjadi pada waktu tertentu.

b. Distribusi yang merata berarti aktivitas terjadi sepanjang waktu.

c. Puncak-puncak tertentu menunjukkan jam-jam sibuk (misalnya saat banyak pengguna bertransaksi).

6. Tampilan Korelasi Fitur



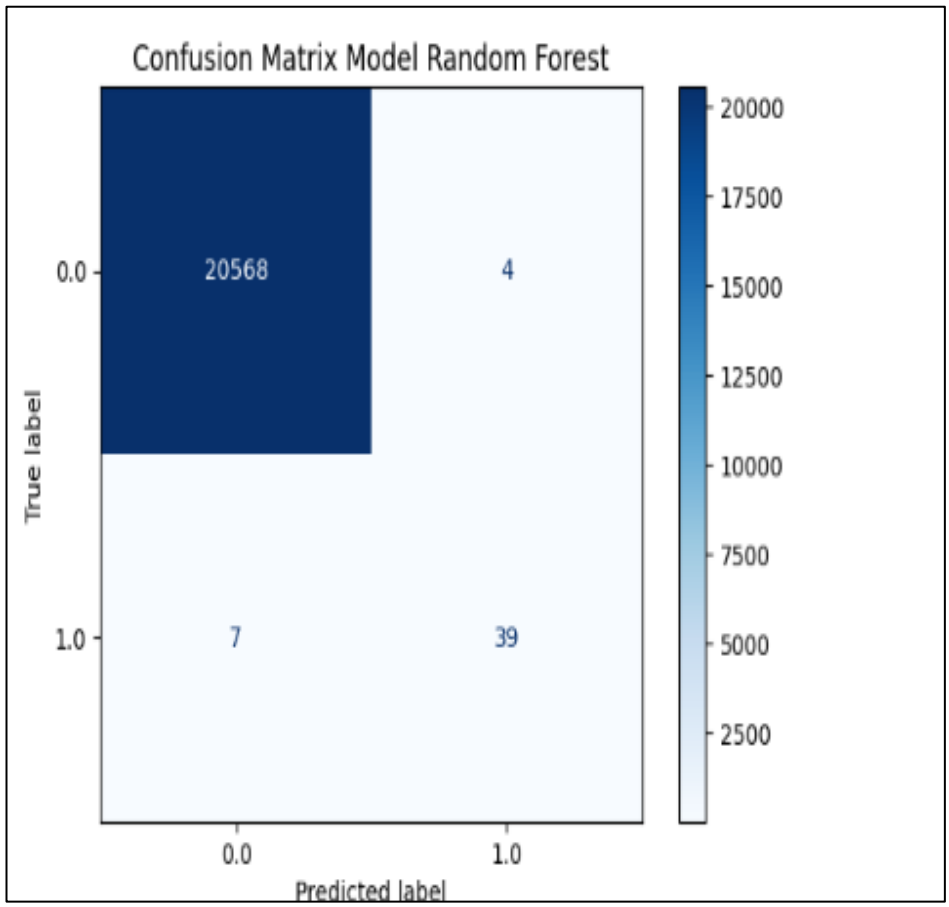
Nilai korelasi positif (mendekati +1) → fitur meningkat seiring dengan kemungkinan fraud.

a. Nilai korelasi negatif (mendekati -1) → fitur menurun seiring dengan kemungkinan fraud.

b. Nilai mendekati 0 → tidak ada hubungan linier yang signifikan.

Heatmap ini membantu feature selection — memilih fitur penting. Fitur dengan korelasi paling besar (positif atau negatif) terhadap Class adalah yang paling informatif bagi model klasifikasi. Meskipun korelasi tinggi berguna, jangan hanya bergantung pada korelasi linier — beberapa fitur bisa memiliki hubungan non-linear yang penting.

7. Tampilan Visualisasi Confusion Matrix

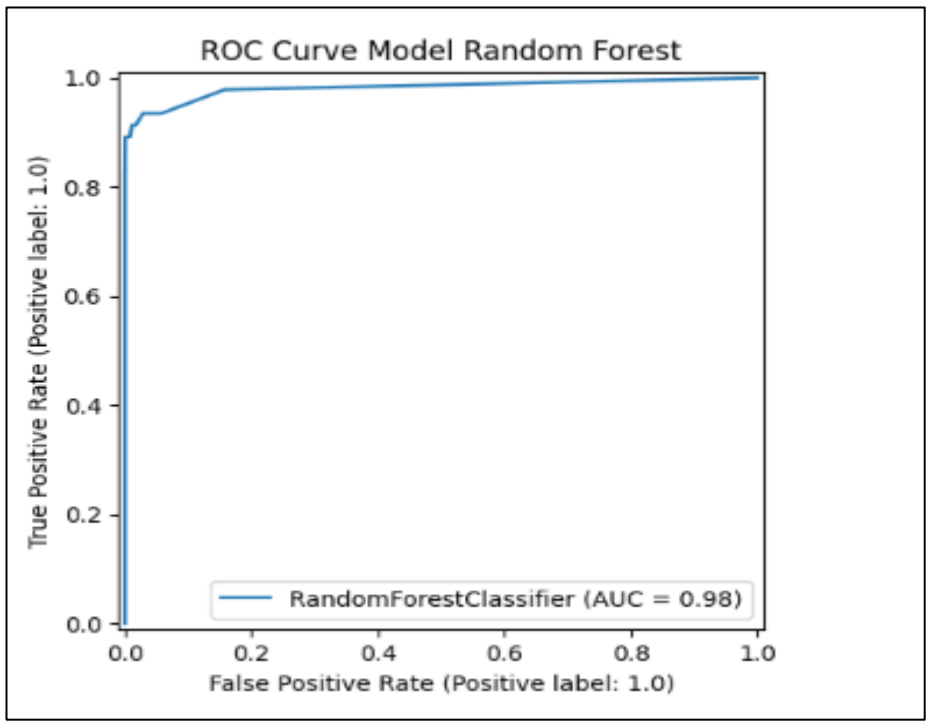


Confusion Matrix biasanya berbentuk 2×2 karena ini adalah masalah klasifikasi biner (fraud vs non-fraud)

a. TN = 20568 → Normal diprediksi normal.
b. FP = 4 → Normal diprediksi fraud (False Alarm).
c. FN = 7 → Fraud diprediksi normal (Kesalahan fatal).
d. TP = 39 → Fraud diprediksi fraud (Deteksi berhasil).

Visualisasi ini membantu memahami kinerja model dalam hal: Akurasi deteksi fraud. Jumlah kesalahan (false positive dan false negative). Semakin banyak angka di diagonal (TN dan TP), semakin baik performa model. False negative (FN) perlu diperhatikan karena artinya fraud tidak terdeteksi, yang berbahaya dalam kasus deteksi penipuan.

8. Tampilan Visualisasi ROC Curve



ROC Curve menunjukkan performa model dalam membedakan dua kelas (fraud dan normal) dengan memplot:

a. True Positive Rate (Recall) pada sumbu Y
b. False Positive Rate (FPR) pada sumbu X

ROC Curve ideal:

- Lengkungannya menyudut ke kiri atas, menandakan tingkat deteksi yang tinggi dengan kesalahan minimum.

Nilai ini mengindikasikan:

a. 0.5 = model tidak lebih baik dari tebak-tebakan.
b. 1.0 = model sempurna.
c. 0.9820 = sangat bagus, model sangat efektif membedakan transaksi fraud dan normal.

ROC Curve memberikan visualisasi penting untuk mengevaluasi kemampuan deteksi model terhadap kelas fraud dibandingkan dengan kelas mayoritas. AUC yang mendekati 1 menunjukkan bahwa model sangat baik dalam klasifikasi, cocok untuk kasus deteksi penipuan.

BAB V
PENUTUP

5.1 Kesimpulan

Dalam proyek klasifikasi transaksi penipuan menggunakan model Random Forest, dilakukan serangkaian tahapan mulai dari preprocessing data, eksplorasi, pelatihan model, hingga evaluasi kinerja. Pada tahap preprocessing, data dibersihkan dari nilai kosong (missing values) khususnya pada kolom target. Kemudian dilakukan normalisasi data menggunakan StandardScaler agar setiap fitur memiliki skala yang seragam, yang penting untuk kinerja algoritma pembelajaran mesin. Mengingat ketidakseimbangan kelas pada dataset — di mana transaksi normal jauh lebih banyak dibanding penipuan — digunakan teknik SMOTE (Synthetic Minority Over-sampling Technique) untuk menyeimbangkan data pelatihan, sehingga model tidak bias terhadap kelas mayoritas.

Selanjutnya, dilakukan eksplorasi data secara visual untuk memahami pola dan distribusi fitur seperti Amount (jumlah transaksi) dan Time (waktu transaksi). Histogram dan heatmap korelasi menunjukkan bahwa sebagian besar fitur memiliki korelasi rendah terhadap target Class, karena data telah melalui proses reduksi dimensi (PCA) sejak awal. Visualisasi distribusi kelas menegaskan bahwa transaksi penipuan merupakan minoritas ekstrem, sehingga pendekatan oversampling sangat penting.

Model Random Forest kemudian dilatih pada data yang telah diresample, menghasilkan performa yang sangat baik. Evaluasi menunjukkan bahwa model mampu meminimalkan kesalahan deteksi, dengan nilai akurasi mendekati 100%, precision dan recall untuk kelas penipuan di atas 85%, serta nilai ROC-AUC sebesar 0.9820 — yang menunjukkan kemampuan model dalam membedakan antara transaksi penipuan dan normal sangat tinggi. Confusion matrix dan kurva ROC yang ditampilkan memperkuat bukti efektivitas model dalam klasifikasi.

Secara keseluruhan, dapat disimpulkan bahwa kombinasi preprocessing yang tepat, eksplorasi data yang mendalam, dan penggunaan model Random Forest dengan penanganan ketidakseimbangan kelas berhasil menciptakan sistem deteksi penipuan yang akurat dan andal. Meskipun demikian, dalam penerapan dunia nyata, penting

untuk terus memantau dan mengevaluasi model secara berkala agar tetap akurat terhadap pola penipuan baru yang mungkin muncul.