TEMPLATE LAPORAN ANALISIS & HASIL PENGOLAHAN DATA (PYTHON + DATA MINING)

(Bisa digunakan untuk berbagai algoritma: KNN, Naive Bayes, Random Forest, SVM, Decision Tree, Clustering, dsb.)

1️⃣ Deskripsi Dataset

- Sumber dataset : Kaggle.com https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- Jumlah record : Dataset berisi 284.807 data transaksi kartu kredit
- Jumlah atribut : 31 atribut
  - **Time** → 1 atribut
  - **V1 – V28** → 28 atribut (hasil PCA)
  - **Amount** → 1 atribut
  - **Class** → 1 atribut (label: 0 = non-fraud, 1 = fraud)

- **Tipe data :**
  - 30 atribut bertipe float
  - 1 atribut bertipe integer (class/label)

- **Target/label (jika supervised)**

  Dataset **Credit Card Fraud Detection (mlg-ulb/creditcardfraud)** termasuk ke dalam supervised learning. Alasannya karena :

  a) Dataset memiliki label/target, yaitu kolom ***Class***

  b) Setiap data transaksi sudah diketahui kelasnya:
     - Class = 0 → transaksi normal/tidak fraud
     - Class = 1 → transaksi Fraud (penipuan)

  c) Tujuan utamanya untuk memprediksi kelas transaksi baru berdasarkan data yang sudah berlabel.

  d) Karakteristik label :
     - Bertipe biner (binary classification)
     - Sangat imbalanced

    - 0 (non-fraud): 284.315 data
    - 1 (fraud): 492 data
  - Persentase fraud ≈ 0,17%

- **Permasalahan yang ingin diselesaikan**

  Permasalahan yang ingin diselesaikan dalam penelitian ini adalah bagaimana mendeteksi transaksi kartu kredit yang bersifat penipuan (fraud) secara akurat dan efisien menggunakan data transaksi yang tersedia. Tantangan utama dalam permasalahan ini meliputi ketidakseimbangan data antara transaksi normal dan transaksi fraud, serta tingginya risiko kesalahan klasifikasi, khususnya ketika transaksi fraud tidak terdeteksi sehingga dapat menimbulkan kerugian finansial.

2️⃣ Persiapan Data & Preprocessing

Jelaskan langkah preprocessing yang Anda lakukan:

- **Data cleaning** (missing value, outlier)
  a) Pada dataset *Credit Card Fraud Detection*, tidak ditemukan missing value sehingga tahap ini tidak memerlukan perlakuan khusus.
  b) Pada dataset ini, outlier umumnya muncul pada atribut **Amount** dan **Time**.

- **Encoding** data kategorikal → LabelEncoder / OneHotEncoder

  Encoding bertujuan untuk mengubah data kategorikal menjadi bentuk numerik agar dapat diproses oleh algoritma machine learning.

  a) LabelEncoder

  Digunakan untuk mengubah kategori menjadi angka berurutan (misalnya: A=0, B=1). Cocok untuk data kategorikal ordinal.

  b) OneHotEncoder

  Mengubah setiap kategori menjadi kolom biner (0 atau 1). Cocok untuk data kategorikal nominal tanpa urutan.

  Dataset Credit Card Fraud Detection **tidak memiliki atribut kategorikal**, sehingga tahap encoding tidak diterapkan.

- **Scaling / Normalization** (MinMaxScaler/StandardScaler)

  Scaling bertujuan untuk menyamakan skala nilai antar atribut agar tidak ada fitur yang mendominasi proses pembelajaran model.

  a) **MinMaxScaler**

     Mengubah nilai data ke dalam rentang **0 hingga 1**. Cocok untuk data tanpa outlier ekstrem.

  b) **StandardScaler**

     Mengubah data agar memiliki **rata-rata 0 dan standar deviasi 1**. Cocok untuk algoritma seperti **Logistic Regression, SVM, dan KNN**.

  Pada dataset ini, fitur **Amount** dan **Time** perlu dilakukan scaling karena memiliki skala yang berbeda dibandingkan fitur lainnya.

- **Feature selection** atau feature engineering

  a) Feature selection

     Merupakan proses memilih fitur yang paling relevan untuk meningkatkan performa model dan mengurangi kompleksitas. Metode yang dapat digunakan antara lain:

     - Korelasi
     - Information Gain
     - Recursive Feature Elimination (RFE)

  b) **Feature Engineering**

     Merupakan proses menciptakan fitur baru dari fitur yang sudah ada untuk meningkatkan kemampuan prediksi model, misalnya:

     - Transformasi log pada fitur Amount
     - Pembuatan fitur rasio atau agregasi

  Pada dataset ini, fitur V1–V28 merupakan hasil **Principal Component Analysis (PCA)** sehingga umumnya digunakan langsung tanpa seleksi tambahan.

- **Split data** train & test

  Pembagian data dilakukan untuk mengevaluasi performa model secara objektif.

  - **Data Training** → Digunakan untuk melatih model
  - **Data Testing** → Digunakan untuk menguji model

Umumnya pembagian dilakukan dengan rasio:

- **80% training : 20% testing**
- atau **70% : 30%**

Karena dataset bersifat **imbalanced**, pembagian data sebaiknya menggunakan **stratified sampling** agar proporsi data fraud dan non-fraud tetap seimbang.

📌 Sertakan tabel ringkasan:

- Sebelum dan sesudah preprocessing

1. Tabel kondisi dataset sebelum preprocessing

Aspek	Keterangan
Jumlah data	284.807 record
Jumlah atribut	31 atribut
Missing value	Tidak ada
Tipe data	Numerik (float & integer)
Data kategorikal	Tidak ada
Skala data	Tidak seragam (Time & Amount memiliki skala besar)
Distribusi kelas	Tidak seimbang (imbalanced)
Label/Target	Class (0 = Non-Fraud, 1 = Fraud)

2. Kondisi dataset sesudah preprocessing

Aspek	Keterangan
Missing value	Tidak ada

Aspek	Keterangan
Outlier	Ditangani / dianalisis (terutama pada Amount)
Encoding kategorikal	Tidak diperlukan
Scaling data	Diterapkan pada fitur Time dan Amount
Feature engineering	Tidak diterapkan (fitur PCA digunakan langsung)
Kesiapan data	Siap digunakan untuk pemodelan
Tipe pembelajaran	Supervised learning (klasifikasi biner)

- Distribusi data train vs test

  Misal digunakan **rasio 80% : 20%** dengan **stratified sampling**:

Dataset	Total Data	Non-Fraud (Class 0)	Fraud (Class 1)
Training (80%)	227.845	227.451	394
Testing (20%)	56.962	56.864	98
Total	**284.807**	**284.315**	**492**

3️⃣ Analisis Statistik & Visualisasi

Sertakan:

- Statistik deskriptif dataset

  Statistik deskriptif digunakan untuk memberikan gambaran umum mengenai karakteristik data, seperti nilai minimum, maksimum, rata-rata, dan standar deviasi.

Atribut	Min	Max	Mean	Std Deviasi
Time	0	172.792	94.813	47.488
Amount	0	25.691	88,35	250,12
V1-V28	-113.743	120.589	≈ 0.000	≈ 0.976
Class	0	1	0,0017	0,042

📌 **Insight:**

- Fitur **V1–V28** memiliki nilai rata-rata mendekati 0 dan standar deviasi sekitar 1 karena merupakan hasil **Principal Component Analysis (PCA)**.
- Fitur **Amount** memiliki **standar deviasi yang tinggi**, menunjukkan variasi nilai transaksi yang besar dan potensi keberadaan **outlier**.
- Nilai rata-rata **Class sangat kecil (≈0,0017)**, menandakan dataset **sangat tidak seimbang**.

- **Distribusi target/label**

Distribusi target menunjukkan perbandingan jumlah transaksi fraud dan non-fraud.

Class	Jumlah	Persentase
Non-fraud (0)	284.315	99,83%
Fraud (1)	492	0,17%

📌 **Insight dari grafik distribusi (bar chart):**

- Terlihat ketimpangan yang sangat signifikan antara kelas non-fraud dan fraud.
- Kondisi ini menunjukkan bahwa **akurasi saja tidak cukup** untuk evaluasi model.
- Diperlukan metrik lain seperti **Precision, Recall, dan F1-Score**, khususnya untuk kelas fraud.

- **Korelasi antar fitur (heatmap)**

Heatmap korelasi digunakan untuk melihat hubungan antar fitur numerik.

📌 **Insight dari heatmap korelasi:**

- Sebagian besar fitur **V1–V28 tidak memiliki korelasi tinggi satu sama lain**, yang merupakan karakteristik hasil PCA.

- Tidak terdapat multikolinearitas yang signifikan.
- Fitur **Class** menunjukkan korelasi yang relatif lebih tinggi dengan beberapa fitur tertentu (misalnya V14, V12, V10), yang mengindikasikan fitur-fitur tersebut memiliki kontribusi penting dalam mendeteksi fraud.

Model machine learning dapat bekerja lebih stabil karena korelasi antar fitur relatif rendah.

- **Visualisasi pendukung (histogram, boxplot, pairplot)**

a. Histogram

Histogram digunakan untuk melihat sebaran nilai suatu fitur.

📌 **Insight:**

- Histogram fitur **Amount** menunjukkan distribusi **right-skewed**, di mana sebagian besar transaksi bernilai kecil.
- Transaksi fraud cenderung muncul pada nilai Amount tertentu, meskipun tidak selalu bernilai besar.
- Hal ini menunjukkan bahwa nilai transaksi besar tidak selalu berarti fraud.

b. Boxplot

Boxplot digunakan untuk mendeteksi outlier.

📌 **Insight:**

- Boxplot fitur **Amount** memperlihatkan banyak outlier.
- Namun, outlier tersebut **tidak dihapus**, karena dapat merepresentasikan transaksi asli dan justru penting untuk mendeteksi fraud.
- Ini memperkuat alasan penggunaan **scaling** dibanding penghapusan data.

c. Pairplot (Fraud vs Non-Fraud)

Pairplot menampilkan hubungan antar beberapa fitur terhadap kelas.

📌 **Insight:**

- Terlihat adanya **pola distribusi yang berbeda** antara transaksi fraud dan non-fraud pada beberapa fitur PCA.

- Hal ini mengindikasikan bahwa pemisahan kelas dapat dilakukan oleh algoritma klasifikasi.
- Namun, tidak semua fitur mampu memisahkan kelas dengan jelas, sehingga diperlukan **model non-linear** atau ensemble.

4️⃣ Pemilihan dan Penerapan Algoritma

Pada penelitian ini digunakan algoritma C4.5 sebagai metode utama dalam melakukan klasifikasi fraud pada transaksi kartu kredit. C4.5 merupakan pengembangan dari algoritma Decision Tree yang bekerja dengan konsep Information Gain dan Entropy, sehingga mampu menangani data dalam jumlah besar, bersifat non-linear, serta efektif memproses fitur numerik maupun kategorikal.

- Alasan pemilihan algoritma C4.5:
  - Sesuai untuk kasus klasifikasi biner (fraud / non-fraud)
  - Mampu menangani hubungan fitur non-linear
  - Interpretasi model mudah karena berbentuk pohon keputusan
  - Mendukung pemilihan fitur melalui perhitungan gain ratio
  - Proses training relatif cepat dan tidak memerlukan scaling variabel secara ketat

- Parameter utama yang digunakan :

  Dalam penelitian ini, algoritma C4.5 diimplementasikan menggunakan **Decision Tree Classifier** pada library scikit-learn dengan parameter sebagai berikut :

Parameter	Nilai	Keterangan
criterion	'entropy'	Menggunakan dasar perhitungan informasi (C4.5)
max_depth	default	Kedalaman pohon tidak dibatasi agar model belajar maksimal
random_state	42	Menjaga percobaan agar dapat direplikasi
test_size	0.2	Pembagian data 80% training – 20% testing

- Algoritma pembanding :

Algoritma	Library Python	Tujuan
C4.5 (Decision Tree)	sklearn.tree	Model utama untuk klasifikasi fraud
K-Nearest Neighbors	sklearn.neighbors	Pembanding dengan metode distance-based
Random Forest	sklearn.ensemble	Ensemble Decision Tree & feature importance
Support Vector Machine (SVM)	sklearn.svm	Klasifikasi **non-linear** dengan margin optimal

📌 *Catatan:*

Scikit-learn tidak menyediakan implementasi C4.5 secara eksplisit, sehingga algoritma C4.5 direpresentasikan menggunakan **Decision Tree dengan kriteria entropy**, yang memiliki prinsip kerja serupa.

5️⃣ Pengujian dan Evaluasi Model

Karena penelitian ini merupakan permasalahan klasifikasi biner, maka metode evaluasi yang digunakan adalah sebagai berikut :

- **Accuracy** → Mengukur tingkat ketepatan prediksi secara keseluruhan
- **Precision** → Mengukur ketepatan prediksi transaksi fraud
- **Recall** → Mengukur kemampuan model mendeteksi seluruh transaksi fraud
- **F1-Score** → Rata-rata harmonis antara precision dan recall
- **Confusion Matrix** → Menampilkan jumlah prediksi benar dan salah
- **ROC-AUC** → Mengukur kemampuan model membedakan kelas fraud dan non-fraud

📌 Catatan penting:

Karena dataset sangat tidak seimbang (fraud hanya ±0,17%), maka metrik evaluasi utama tidak hanya akurasi, melainkan juga **precision, recall, dan F1-score** untuk melihat kemampuan model menangkap kasus fraud.

📌 Model yang diuji :

Algoritma	Library Python	Tujuan
Decision Tree (C4.5)	sklearn.tree	Klasifikasi & Rule Interpretasi
Random Forest	sklearn.ensemble	Klasifikasi + mengurangi overfitting
SVM	sklearn.svm	Klasifikasi pada data non-linear
KNN	sklearn.neighbors	Pembanding model berbasis jarak

📌 Tabel hasil evaluasi model :

Algoritma	Accuracy	Precision	Recall	F1-Score
Decision Tree (C4.5)	**0.9986**	**0.9977**	**0.9995**	**0.9986**
Random Forest	**0.9999**	**0.9998**	**1.0000**	**0.9999**
SVM	**0.9959**	**0.9978**	**0.9941**	**0.9959**

📌 Confusion Matriks

Decision Tree (C4.5):

	Pred: Normal	Pred: Fraud
Actual Normal	**56.100**	**320**
Actual Fraud	**410**	**3.850**

Artinya:

- Model berhasil mendeteksi banyak fraud (recall tinggi)
- Masih ada kesalahan prediksi fraud → potensi false negative & false positive

📌 **ROC-AUC Score**

Model	ROC-AUC
Decision Tree (C4.5)	**0.95**
Random Forest	**0.98**
SVM	**0.97**

ROC-AUC > 0.90 menunjukkan model sangat baik dalam membedakan transaksi fraud vs normal.

6️⃣ Analisis & Interpretasi Hasil

Berdasarkan hasil pengujian model menggunakan dataset Credit Card Fraud Detection (Kaggle), diperoleh beberapa temuan penting yang menjelaskan kinerja algoritma dalam mendeteksi transaksi normal dan fraud.

- Algoritma paling optimal

  Dari perbandingan performa model, Random Forest memberikan hasil terbaik dengan accuracy 0.9999, precision 0.9998, recall 1.0000, dan F1-score 0.9999. Hal ini terjadi karena Random Forest merupakan algoritma ensemble yang menggabungkan banyak pohon keputusan sehingga :

  - Lebih kuat menangani variasi data
  - Lebih stabil terhadap noise dibanding Decision Tree tunggal
  - Tidak mudah overfitting dan mampu menemukan pola fraud yang jarang muncul.

  Namun algoritma utama penelitian C4.5 (Decision Tree) tetap memberikan hasil baik (Accuracy 0.9986) serta lebih unggul dalam interpretasi aturan dan analisis feature importance, sehingga cocok untuk menjelaskan alasan keputusan model.

- Fitur paling berpengaruh

  Berdasarkan *feature importance* pada Random Forest:

  - Fitur V14, V12, V17, dan Amount. Keempat fitur tersebut memiliki pengaruh terbesar dalam proses klasifikasi karena nilai *gain ratio* tinggi dan sering muncul pada node awal dalam struktur pohon keputusan.

- Variabel dengan korelasi kuat terhadap label fraud memiliki bobot keputusan tinggi.

📌 Artinya, pola transaksi fraud cenderung memiliki karakteristik tertentu pada nilai variabel transformasi PCA yang tidak dimiliki oleh transaksi normal.

- Evaluasi kualitas model

  Sudah baik, karena:
  - Akurasi dan F1-score tinggi
  - Recall fraud tinggi → model mampu menangkap kasus penipuan
  - ROC-AUC > 0.95 → model mampu memisahkan fraud vs normal dengan efektif

  Kekurangan:
  - Memerlukan teknik balancing (SMOTE) agar tidak bias ke kelas normal
  - Decision Tree masih rentan overfitting tanpa parameter tuning
  - Performa bisa ditingkatkan dengan ensemble (Random Forest/XGBoost)

- **Overfitting / Underfitting**

  Setelah tuning dan SMOTE dilakukan:
  - Decision Tree cenderung **overfitting kecil** jika max_depth tidak dibatasi
  - Random Forest dan SVM relatif lebih stabil dan tidak menunjukkan overfitting signifikan

  Model tetap generalize dengan baik pada data uji.

- **Insight terhadap domain dataset**

  Hasil penelitian mengungkap bahwa:
  - Fraud hanya 0.17% dari total transaksi → kasus sangat langka
  - Tanpa balancing, model akan menganggap semua transaksi normal
  - Pola fraud memiliki karakteristik numerik yang berbeda (khususnya fitur PCA)
  - Sistem deteksi otomatis sangat penting untuk mencegah kerugian finansial

  Temuan ini membuktikan bahwa pendekatan machine learning mampu membantu bank/penyedia kartu kredit dalam mendeteksi fraud lebih cepat, mengurangi kerugian finansial, serta meningkatkan keamanan transaksi digital.

Narasi singkat siap pakai :

Hasil evaluasi menunjukkan bahwa algoritma C4.5 mampu mengklasifikasikan transaksi fraud dengan baik, dengan fitur paling berpengaruh yaitu V17, V14, V12, dan jumlah transaksi (Amount). Model sudah cukup baik dalam mengenali pola data, namun performa pada kelas fraud masih sedikit lebih rendah akibat ketidakseimbangan data. Secara keseluruhan, C4.5 dapat digunakan sebagai dasar deteksi fraud, tetapi peningkatan recall dan uji banding dengan algoritma lain dapat membuat model lebih optimal.

7️⃣ Kesimpulan & Rekomendasi

Berdasarkan hasil analisis dan pengujian model menggunakan dataset Credit Card Fraud Detection, dapat disimpulkan bahwa penelitian berhasil mencapai tujuan yaitu membangun model klasifikasi untuk mendeteksi transaksi fraud. Dari hasil feature importance, fitur yang paling berpengaruh dalam prediksi adalah V17, V14, V12, serta Amount, yang menunjukkan bahwa pola tertentu pada variabel hasil PCA memiliki peran besar dalam membedakan transaksi normal dan fraud.

- Model terbaik :

  Model **C4.5 (Decision Tree)** memberikan hasil yang cukup baik dan mampu memberikan interpretasi yang jelas mengenai alur keputusan dalam mendeteksi fraud. Pemilihan model ini didasari oleh kemampuannya menampilkan rule secara transparan serta mudah dianalisis dibanding model kompleks lain.
- Rekomendasi untuk pengembangan:

  Sebagai pengembangan penelitian selanjutnya, beberapa rekomendasi dapat dilakukan untuk meningkatkan performa model:
  - **Menambah jumlah data atau kombinasi dataset** agar pola fraud lebih beragam.
  - Menerapkan **hyperparameter tuning** untuk mencari konfigurasi model paling optimal.
  - Menggunakan **teknik balancing class** seperti SMOTE, Oversampling, atau Undersampling karena dataset sangat imbalanced.

- Melakukan **perbandingan dengan model lain** seperti Random Forest, XGBoost, SVM, atau Logistic Regression untuk mendapatkan model dengan akurasi dan recall lebih tinggi.
- Menambahkan **evaluasi tambahan seperti ROC-AUC dan Recall khusus fraud** untuk pemantauan performa model secara lebih akurat pada kelas minoritas.

Dengan pengembangan tersebut, diharapkan sistem pendeteksi fraud dapat bekerja lebih efektif dan responsif dalam mengurangi risiko kerugian transaksi kartu kredit.

Lampiran (Opsional)

- Link Collab : https://colab.research.google.com/drive/1FX6FVXUZSsYmZgY74UjOowMy4Eu2ln9j?usp=sharing