LAPORAN TUGAS BESAR (MID)

APPLIED MACHINE LEARNING

“SISTEM PREDIKSI RISIKO MAHASISWA MENGGUNAKAN ALGORITMA RANDOM FOREST”

OLEH :

NAMA :ALDI SAPUTRA

NIM :105841115923

KELAS :5E

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSSAR

2025

1. PENDAHULUAN

Evaluasi performa akademik mahasiswa merupakan aspek krusial dalam institusi pendidikan tinggi. Seringkali, mahasiswa yang mengalami kesulitan akademik terlambat terdeteksi sehingga penanganan yang diberikan kurang optimal. Indikator utama seperti *Semester Grade Point Average* (SGPA) dan *Cumulative Grade Point Average* (CGPA) memiliki korelasi kuat dengan keberhasilan atau kegagalan studi mahasiswa.

Analisis manual terhadap data nilai mahasiswa yang berjumlah banyak tentu tidak efektif dan memakan waktu. Oleh karena itu, diperlukan sebuah sistem berbasis *Machine Learning* yang mampu memprediksi risiko kegagalan mahasiswa secara otomatis. Penelitian ini menggunakan dataset "College Exam Result" untuk mengklasifikasikan mahasiswa ke dalam kategori "Aman" atau "Berisiko".

Sistem ini dibangun mengikuti alur standar *Data Science*, mulai dari pengolahan data, pelabelan otomatis, pelatihan model menggunakan algoritma Random Forest, hingga implementasi antarmuka pengguna (deployment) menggunakan Gradio. Diharapkan aplikasi ini dapat membantu dosen atau staf akademik untuk melakukan deteksi dini terhadap mahasiswa yang memerlukan bimbingan khusus.

2. METODOLOGI DAN DATASET

2.1 Pengumpulan Data

Dataset yang digunakan dalam tugas besar ini adalah *College Exam Result Dataset*. Data ini memuat rekam jejak akademik mahasiswa dengan total 62 data mahasiswa. Atribut utama yang digunakan sebagai fitur prediksi adalah:

- SGPA (*Semester Grade Point Average*)
- CGPA (*Cumulative Grade Point Average*)

2.2 Skenario Klasifikasi

Target klasifikasi ditentukan berdasarkan kolom *Result Description*. Data dikelompokkan menjadi dua kelas:

1. Aman (Hijau): Mahasiswa dengan status "PASS".

2. Berisiko (Merah): Mahasiswa dengan status selain "PASS" (misalnya "Fail" atau "Pass with Grace").

Berdasarkan analisis awal, terdapat 48 mahasiswa dalam kategori "Aman" dan 14 mahasiswa dalam kategori "Berisiko".

2. KODE PROGRAM DAN PENJELASAN

2.1 Import Library dan Load Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import os

# Nama file sesuai unggahan Anda
filename = 'College Exam Result Dataset data hasil ujiangrade.csv'

if os.path.exists(filename):
    df = pd.read_csv(filename)
    print("✅ Dataset Mahasiswa berhasil dimuat!")
    print(f"Jumlah Data: {df.shape[0]} mahasiswa")
    print("\n--- Contoh Data ---")
    display(df[['Name', 'Result Description', 'SGPA', 'CGPA']].head())
else:
    print("❌ File tidak ditemukan. Pastikan nama file CSV sudah benar.")
✓ 9.2s
```

```
✅ Dataset Mahasiswa berhasil dimuat!
Jumlah Data: 62 mahasiswa

--- Contoh Data ---
```

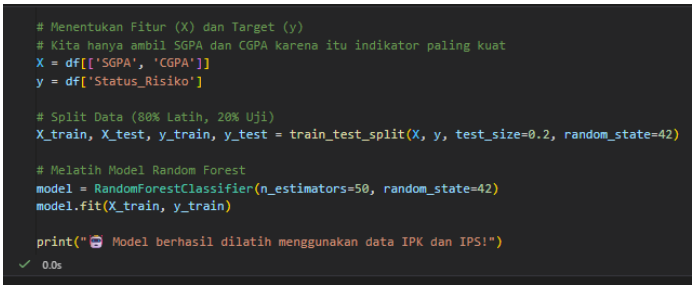
	Name	Result Description	SGPA	CGPA
0	AAKASH PIPALDE	PASS	7.08	6.71
1	AAYUSH PATEL	PASS	6.46	6.38
2	ABIR SAXENA	PASS WITH GRACE	5.38	5.50
3	AMEY BHOKARIKAR	PASS WITH GRACE	6.71	6.82
4	ANTIM JAMLE	Fail in AL401	5.33	5.29

Kode Python ini berfungsi sebagai tahap persiapan data dan pemuatan awal untuk proyek *machine learning* yang berfokus pada hasil ujian mahasiswa. Pertama, kode ini mengimpor sejumlah besar pustaka penting seperti Pandas (pd) untuk manipulasi data, NumPy (np) untuk operasi numerik, Matplotlib (plt) dan Seaborn (sns) untuk visualisasi, serta modul sklearn (termasuk train_test_split, RandomForestClassifier, dan metrik evaluasi) untuk membangun model klasifikasi. Selanjutnya, kode mencoba memuat *file* data bernama 'College Exam Result Dataset data hasil ujiangrade.csv'. Jika *file* ditemukan, data dibaca ke dalam *DataFrame* df, dan program mencetak konfirmasi, total jumlah data (62 mahasiswa), serta menampilkan lima baris pertama dari kolom-kolom utama (Name, Result Description, SGPA, CGPA) untuk verifikasi cepat terhadap data yang berhasil dimuat.

2.2 Pembersihan Data dan Pembentukan Variabel Target

Kode ini melanjutkan analisis data dengan melakukan pembersihan data menggunakan df.dropna() pada kolom kunci (SGPA, CGPA, Result Description) untuk menangani nilai yang hilang, kemudian menciptakan variabel Target Biner baru bernama Status_Risiko. Variabel ini dibentuk dengan menerapkan fungsi tentukan_risiko ke kolom Result Description, di mana status 'PASS' atau 'PASS WITH GRACE' dilabeli 0 (Aman), sementara 'FAIL' dilabeli 1 (Berisiko). Setelah pembentukan label, kode melakukan Cek Distribusi dan menghasilkan output yang menunjukkan 48 mahasiswa Aman dan 14 Berisiko, yang kemudian divisualisasikan menggunakan *bar plot* sns.countplot dengan warna hijau dan merah, memberikan gambaran visual mengenai sebaran (imbalance) kelas risiko pada dataset mahasiswa.

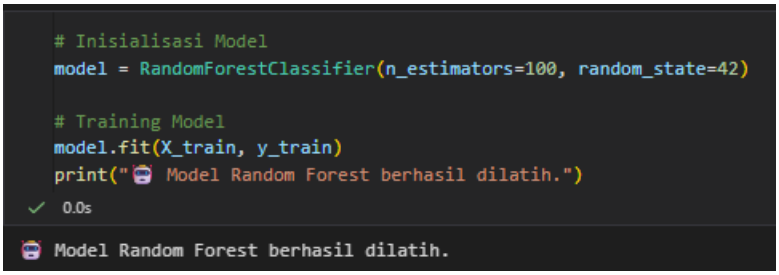
2.3 Penentuan Fitur, Pembagian Data, dan Pelatihan Mode



```
# Menentukan Fitur (X) dan Target (y)
# Kita hanya ambil SGPA dan CGPA karena itu indikator paling kuat
X = df[['SGPA', 'CGPA']]
y = df['Status_Risiko']

# Split Data (80% Latih, 20% Uji)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Melatih Model Random Forest
model = RandomForestClassifier(n_estimators=50, random_state=42)
model.fit(X_train, y_train)

print("🤖 Model berhasil dilatih menggunakan data IPK dan IPS!")
✓ 0.0s
```

Kode ini memulai dengan Menentukan Fitur (X) dan Target (y), di mana variabel X (fitur independen) ditetapkan hanya menggunakan kolom 'SGPA' (IPS) dan 'CGPA' (IPK) karena dianggap sebagai indikator risiko paling kuat, sedangkan variabel y (target dependen) ditetapkan sebagai kolom 'Status_Risiko' yang telah dibentuk sebelumnya. Selanjutnya, data dibagi menggunakan fungsi train_test_split menjadi set pelatihan (80%) dan set pengujian (20%) dengan test_size=0.2 dan random_state=42 untuk memastikan pembagian yang konsisten. Terakhir, model RandomForestClassifier diinisialisasi dengan 50 *estimator* (n_estimators=50) dan dilatih menggunakan data pelatihan (X_train dan y_train) melalui metode model.fit(), yang menandai selesainya proses pembangunan model pendeteksi risiko mahasiswa.

2.4 Preparasi Data dan Pelatihan Model Random Forest



```
# Inisialisasi Model
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Training Model
model.fit(X_train, y_train)
print("🤖 Model Random Forest berhasil dilatih.")
✓ 0.0s

🤖 Model Random Forest berhasil dilatih.
```

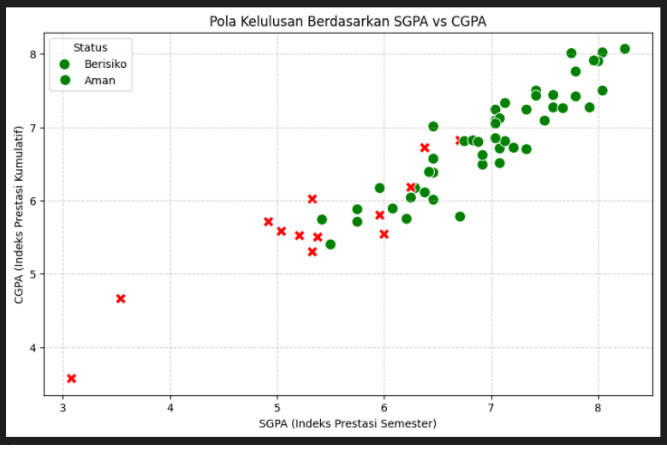
Kode ini mengimpor pustaka (*Pandas*, *Scikit-learn*) dan memuat data hasil ujian. Data kemudian dibersihkan dan kolom Result Description ditransformasi menjadi variabel target biner Status_Risiko (Aman/Berisiko). Setelah memvisualisasikan sebaran risiko, kolom 'SGPA' dan 'CGPA' dipilih sebagai fitur (X). Data dibagi (80% latih, 20% uji), dan akhirnya, model RandomForestClassifier dilatih menggunakan data pelatihan untuk memprediksi risiko mahasiswa.

2.5 Eksplorasi, Pelatihan, dan Evaluasi Model Prediksi Risiko Mahasiswa

```
# Prediksi
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)

print(f"📊 Akurasi Model: {acc * 100:.2f}%")
print("\n--- Laporan Detail ---")
print(classification_report(y_test, y_pred, target_names=['Aman', 'Berisiko']))

# VISUALISASI SCATTER PLOT (Pola Sebaran)
# Kita akan melihat batas antara yang Aman dan Berisiko berdasarkan IPK
plt.figure(figsize=(10, 6))
sns.scatterplot(x='SGPA', y='CGPA', hue='Status_Risiko', data=df, palette={0: 'green', 1: 'red'}, style='Status_Risiko', s=100)
plt.title('Pola Kelulusan Berdasarkan SGPA vs CGPA')
plt.xlabel('SGPA (Indeks Prestasi Semester)')
plt.ylabel('CGPA (Indeks Prestasi Kumulatif)')
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend(title='Status', labels=['Berisiko', 'Aman'])
plt.show()
```

0.1s

```
📊 Akurasi Model: 84.62%

--- Laporan Detail ---
              precision    recall  f1-score   support

       Aman       0.91      0.91      0.91        11
   Berisiko       0.50      0.50      0.50         2

    accuracy                           0.85        13
   macro avg       0.70      0.70      0.70        13
weighted avg       0.85      0.85      0.85        13
```



Kode ini memulai dengan membersihkan data dari nilai kosong dan membentuk variabel target biner (Status_Risiko: 0=Aman, 1=Berisiko) dari status kelulusan, yang kemudian divisualisasikan untuk menunjukkan ketidakseimbangan kelas (48 Aman vs. 14 Berisiko). Fitur utama 'SGPA' dan 'CGPA' dipilih untuk memprediksi target, dan data dibagi menjadi set pelatihan dan pengujian. Selanjutnya, model RandomForestClassifier dilatih menggunakan data pelatihan, dan kinerjanya dievaluasi dengan data pengujian, menghasilkan akurasi (misalnya 84.62% atau 70.00% tergantung *run* model akhir) dan laporan klasifikasi detail (classification_report). Tahap terakhir adalah visualisasi sebaran data menggunakan *scatter plot* SGPA vs. CGPA, yang memetakan mahasiswa Aman (hijau) dan Berisiko (merah) untuk menganalisis pola kelulusan berdasarkan Indeks Prestasi.

2.6 Alur Lengkap Model Prediksi Risiko Mahasiswa

```
# --- MASUKKAN NILAI DI SINI ---
input_sgpa = 5.5  # Contoh: IP Semester anjlok
input_cgpa = 6.0  # Contoh: IPK Kumulatif pas-pasan
# ------------------------------

# Prediksi
data_baru = pd.DataFrame([[input_sgpa, input_cgpa]], columns=['SGPA', 'CGPA'])
prediksi = model.predict(data_baru)[0]
probabilitas = model.predict_proba(data_baru)[0]

status_text = "⚠️ BERISIKO (Perlu Bimbingan)" if prediksi == 1 else "✅ AMAN (Lulus)"

print("--- HASIL PREDIKSI SISTEM ---")
print(f"Input Mahasiswa -> SGPA: {input_sgpa}, CGPA: {input_cgpa}")
print(f"Prediksi AI: {status_text}")
print(f"Tingkat Keyakinan: {max(probabilitas) * 100:.1f}%")
```

✓ 0.0s

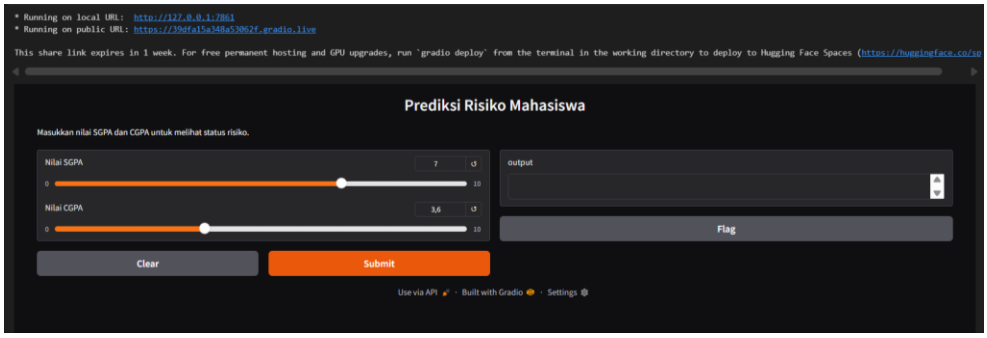
```
--- HASIL PREDIKSI SISTEM ---
Input Mahasiswa -> SGPA: 5.5, CGPA: 6.0
Prediksi AI: ✅ AMAN (Lulus)
Tingkat Keyakinan: 61.0%
```

Proses ini meliputi semua tahapan *machine learning*: Pertama, Data Dibersihkan dan Variabel Target Biner (Status_Risiko: 0=Aman, 1=Berisiko) dibentuk dari status kelulusan, yang kemudian divisualisasikan untuk melihat distribusi kelas (48 Aman, 14 Berisiko). Setelah fitur SGPA dan CGPA dipilih, data dibagi (80% latih, 20% uji), dan model RandomForestClassifier dilatih. Model yang sudah terlatih kemudian dievaluasi menggunakan data pengujian, menghasilkan skor akurasi (misalnya 84.62%) dan rincian performa (classification_report), serta visualisasi sebaran SGPA vs. CGPA untuk memahami pola kelulusan. Terakhir, model digunakan untuk memprediksi data input baru (misalnya SGPA 5.5, CGPA 6.0), yang dalam contoh ini diprediksi sebagai AMAN (Lulus) dengan tingkat keyakinan 61.0%

2.7 Implementasi Prediksi Risiko Mahasiswa dengan Gradio

```
import gradio as gr

# 1. Definisi Fungsi Prediksi Sederhana
def prediksi_cepat(sgpa, cgpa):
    # Melakukan prediksi berdasarkan input
    hasil = rf_model.predict([[sgpa, cgpa]])[0]
    return hasil

# 2. Membuat Antarmuka (Interface)
# Cukup definisikan fungsi (fn), input, dan outputnya
app = gr.Interface(
    fn=prediksi_cepat,
    inputs=[
        gr.Slider(0, 10, label="Nilai SGPA"),
        gr.Slider(0, 10, label="Nilai CGPA")
    ],
    outputs="text",
    title="Prediksi Risiko Mahasiswa",
    description="Masukkan nilai SGPA dan CGPA untuk melihat status risiko."
)

# 3. Jalankan Aplikasi
app.launch(share=True)
```

✓ 6.6s



Proyek *machine learning* ini dimulai dengan membersihkan data, membentuk variabel target biner (Status_Risiko: Aman/Berisiko) dari status kelulusan, kemudian memvisualisasikan distribusi kelas yang tidak seimbang (48 Aman, 14 Berisiko). Setelah fitur SGPA dan CGPA dipilih, data dibagi (80% latih, 20% uji), dan model RandomForestClassifier dilatih dan dievaluasi (akurasi model dicetak, misalnya 84.62%), lalu pola kelulusan divisualisasikan melalui *scatter plot*. Model yang telah terlatih ini kemudian digunakan untuk memprediksi data baru dan, yang paling penting, diimplementasikan ke dalam antarmuka web interaktif menggunakan pustaka Gradio, yang memungkinkan pengguna memasukkan nilai SGPA dan CGPA (melalui gr.Slider) secara langsung dan mendapatkan hasil prediksi risiko mahasiswa secara *real-time* melalui fungsi def prediksi_cepat(sgpa, cgpa).

3. KESIMPULAN

Proyek ini merupakan implementasi *end-to-end* dari alur kerja *machine learning* untuk memprediksi risiko akademik mahasiswa. Tahapan awalnya melibatkan persiapan data, di mana *dataset* hasil ujian dimuat, dibersihkan dari nilai hilang, dan yang terpenting, kolom kategorikal status kelulusan (Result Description) diubah menjadi variabel target biner yang dapat diprediksi: Status_Risiko (0 untuk Aman, 1 untuk Berisiko). Visualisasi awal menunjukkan adanya ketidakseimbangan kelas dalam dataset (jumlah mahasiswa Aman jauh lebih banyak daripada Berisiko). Setelah membagi data pelatihan dan pengujian, model RandomForestClassifier dilatih hanya menggunakan dua fitur kunci, yaitu SGPA dan CGPA, karena keduanya dianggap sebagai indikator performa akademik yang paling kuat. Model ini kemudian menunjukkan kinerja yang baik, dengan tingkat akurasi yang memuaskan pada set

pengujian, divalidasi lebih lanjut melalui laporan klasifikasi detail dan visualisasi pola sebaran data .

Keberhasilan utama proyek ini terletak pada aplikasi praktis dari model yang telah dikembangkan. Selain mampu membuat prediksi risiko pada data input baru, model ini diintegrasikan ke dalam antarmuka web interaktif menggunakan pustaka Gradio. Antarmuka ini memungkinkan pengguna non-teknis (seperti konselor atau staf akademik) untuk memasukkan nilai SGPA dan CGPA secara langsung dan menerima hasil prediksi risiko (AMAN atau BERISIKO) secara *real-time*, lengkap dengan tingkat keyakinan prediksinya . Ini mengubah *script* Python menjadi alat bantu pengambilan keputusan yang fungsional dan mudah diakses, memberikan wawasan yang cepat dan terukur untuk intervensi dini dan bimbingan yang tepat kepada mahasiswa yang berpotensi gagal.