LEMBAR KERJA KERJA MAHASISWA (LKM)

LK.8 Perancangan Project Data Science

Nama	**: RAFLI NAUFAL**
Tanggal	**:6 DESEMBER**
Kelas	**: 5AI-A**
Judul Project : "Deteksi Dini Siswa Berisiko Mendapat Nilai Rendah Menggunakan Analisis Data dan Machine Learning pada Dataset Students Performance"	

1. PENDAHULUAN

1.1 LATAR BELAKANG

Deteksi dini siswa yang berisiko mendapatkan nilai rendah penting dilakukan agar sekolah dapat memberikan intervensi tepat waktu. Sering kali siswa yang mengalami kesulitan belajar terlambat teridentifikasi karena proses pemantauan masih bersifat manual. Dengan memanfaatkan analisis data dan machine learning, sekolah dapat mengetahui pola belajar siswa dan faktor-faktor yang memengaruhi prestasi mereka secara lebih akurat. Dataset *Students Performance* dari Kaggle menyediakan informasi lengkap mengenai karakteristik siswa dan nilai akademik mereka, sehingga sangat cocok digunakan untuk membangun model prediksi yang dapat membantu mendeteksi siswa yang berpotensi memperoleh nilai rendah.

1.2 Dataset yang Digunakan

Dataset yang digunakan adalah StudentsPerformance.csv dari Kaggle yang berisi 1.000 data siswa dengan beberapa fitur utama, seperti jenis kelamin, ras/etnis, tingkat pendidikan orang tua, jenis makan siang, dan keikutsertaan dalam kursus persiapan ujian. Selain itu, dataset ini juga memuat tiga nilai akademik yaitu nilai matematika, membaca, dan menulis. Data tersebut akan dianalisis untuk mengetahui faktor yang memengaruhi prestasi siswa dan digunakan sebagai dasar untuk membangun model machine learning dalam mendeteksi siswa berisiko nilai rendah.

2. METODOLOGI CRISP DM

2.1BUSINESS UNDERSTANDING

Dalam konteks pendidikan, sekolah sering kesulitan mengetahui siswa yang berpotensi memperoleh nilai rendah sebelum hasil akhir diumumkan. Ketika nilai rendah baru diketahui setelah ujian, intervensi biasanya terlambat dan siswa semakin tertinggal. Oleh karena itu, dibutuhkan sistem prediktif yang mampu mendeteksi risiko nilai rendah sejak awal. Data science dan machine learning dapat digunakan untuk mengidentifikasi pola pada data siswa dan memprediksi apakah seorang siswa termasuk kategori berisiko. Tujuan dari proyek ini adalah membangun model machine learning yang mampu melakukan klasifikasi risiko nilai rendah pada siswa berdasarkan data yang tersedia. Model ini nantinya dapat membantu sekolah untuk memberikan intervensi tepat waktu seperti bimbingan tambahan, remedial, atau evaluasi metode belajar

2.2 DATA UNDERSTANDING

Dataset yang digunakan adalah Students Performance yang diperoleh dari Kaggle. Dataset ini memuat informasi nilai akademik siswa dan faktor pendukung lainnya yang relevan dengan prestasi belajar. Pemahaman terhadap struktur dan karakteristik data dilakukan sebelum masuk ke tahap pengolahan lebih lanjut.

```
# Load dataset
df = pd.read_csv("StudentsPerformance.csv")

# Melihat struktur data
df.head()
```

✓ 0.0s

Dari hasil pemeriksaan, diketahui bahwa dataset terdiri dari data numerik (nilai matematika, membaca, dan menulis) serta data kategorikal (gender, pendidikan orang tua, jenis makan siang, dan kursus persiapan). Dataset ini tidak memiliki data teks bebas maupun data berbasis waktu.Target klasifikasi belum tersedia secara langsung, sehingga perlu dibentuk pada tahap berikutnya berdasarkan nilai akademik siswa

2.3 DATA PREPARATION (PERSIAPA DATA)

Tahap ini bertujuan untuk menyiapkan data agar layak digunakan dalam proses pemodelan. Langkah pertama adalah membersihkan data dari duplikasi dan memastikan tidak ada nilai kosong yang dapat memengaruhi hasil analisis.

```
# Menghapus data duplikat
df = df.drop_duplicates()

# Mengecek missing value
df.isnull().sum()
```

✓ 0.0s

Hasil pemeriksaan menunjukkan bahwa dataset tidak memiliki nilai kosong sehingga tidak diperlukan proses imputasi.

Selanjutnya, dibuat variabel target dengan menghitung rata-rata nilai dari tiga mata pelajaran utama. Siswa dengan rata-rata nilai di bawah 60 dikategorikan sebagai siswa berisikonilairendah.

```
# Membuat kolom rata-rata nilai
df['average_score'] = df[['math score', 'reading score', 'writing score']].mean(axis=1)

# Membuat target klasifikasi
df['risk'] = df['average_score'].apply(lambda x: 1 if x < 60 else 0)

df[['average_score', 'risk']].head()
```

✓ 0.0s

Setelah target terbentuk, data kategorikal diubah menjadi numerik menggunakan Label Encoding, kemudian seluruh fitur dinormalisasi agar memiliki skala yang seragam.

```
le = LabelEncoder()

categorical_cols = df.select_dtypes(include='object').columns

for col in categorical_cols:
    df[col] = le.fit_transform(df[col])
```

✓ 0.0s

2.4 MODELING

Pada tahap pemodelan, digunakan dua algoritma klasifikasi, yaitu **K-Nearest Neighbor (KNN)** dan **Naive Bayes**. KNN dipilih karena mampu mengklasifikasikan data berdasarkan

kedekatan jarak antar data, sedangkan Naive Bayes dipilih karena sederhana, cepat, dan efektif pada permasalahan klasifikasi.

```
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)
```

0.0s

```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred_knn = knn.predict(X_test)
```

0.0s

```
nb = GaussianNB()
nb.fit(X_train, y_train)

y_pred_nb = nb.predict(X_test)
```

0.0s

Model dilatih menggunakan data latih dan siap untuk dievaluasi menggunakan data uji.

2.5EVALUTION

Evaluasi dilakukan untuk mengukur performa model dalam memprediksi siswa berisiko nilai rendah. Metode evaluasi yang digunakan adalah accuracy, confusion matrix, dan classification report.

```
print("=== Evaluasi KNN ===")
print("Accuracy:", accuracy_score(y_test, y_pred_knn))
print(confusion_matrix(y_test, y_pred_knn))
print(classification_report(y_test, y_pred_knn))
```

0.0s

```
print("=== Evaluasi Naive Bayes ===")
print("Accuracy:", accuracy_score(y_test, y_pred_nb))
print(confusion_matrix(y_test, y_pred_nb))
print(classification_report(y_test, y_pred_nb))
```

0.0s

Hasil evaluasi ini digunakan untuk membandingkan kinerja kedua model dan menentukan model yang paling sesuai untuk digunakan pada tahap deployment.

Output :

```
=== Evaluasi KNN ===
Accuracy: 0.935
[[135   3]
 [ 10  52]]
              precision    recall  f1-score   support

           0       0.93      0.98      0.95       138
           1       0.95      0.84      0.89        62

    accuracy                           0.94       200
   macro avg       0.94      0.91      0.92       200
weighted avg       0.94      0.94      0.93       200
```

```
=== Evaluasi Naive Bayes ===
Accuracy: 0.95
[[129   9]
 [  1  61]]
              precision    recall  f1-score   support

           0       0.99      0.93      0.96       138
           1       0.87      0.98      0.92        62

    accuracy                           0.95       200
   macro avg       0.93      0.96      0.94       200
weighted avg       0.95      0.95      0.95       200
```

2.6DEPLOYMENT

Pada tahap deployment, model yang telah dilatih disimpan dalam bentuk file .pkl agar dapat digunakan kembali tanpa perlu pelatihan ulang. Penyimpanan model ini penting sebagai bagian dari implementasi sistem berbasis machine learning.

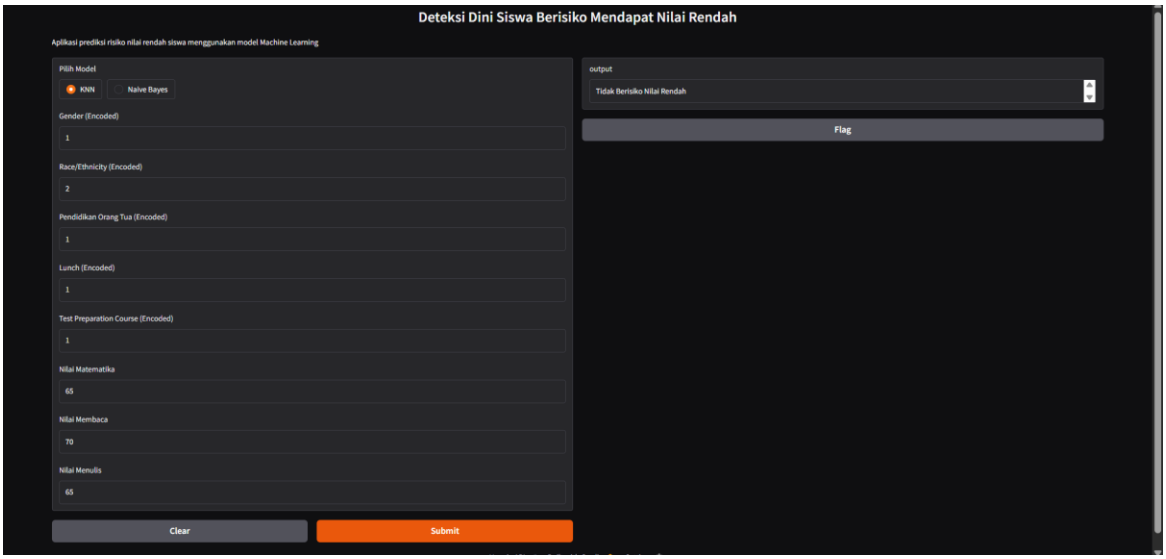
```python
# Simpan model KNN
with open('/mnt/data/model_knn.pkl', 'wb') as f:
    pickle.dump(knn, f)

# Simpan model Naive Bayes
with open('/mnt/data/model_nb.pkl', 'wb') as f:
    pickle.dump(nb, f)

print("Model berhasil disimpan:")
print("- model_knn.pkl")
print("- model_nb.pkl")
```

Model yang telah disimpan selanjutnya dapat diintegrasikan ke dalam aplikasi berbasis web menggunakan **Gradio** sebagai antarmuka pengguna, sehingga sistem dapat digunakan secara langsung oleh pengguna non-teknis.

Berikut adalah kode lengkap antarmuka Gradio:

```python
import gradio as gr
import numpy as np
import pickle

# Load model
with open('/mnt/data/model_knn.pkl', 'rb') as f:
    model_knn = pickle.load(f)

with open('/mnt/data/model_nb.pkl', 'rb') as f:
    model_nb = pickle.load(f)

def predict_risk(
    model_type,
    gender,
    race,
    parental_edu,
    lunch,
    prep_course,
    math_score,
    reading_score,
    writing_score
):
    data = np.array([
        gender,
        race,
        parental_edu,
        lunch,
        prep_course,
        math_score,
        reading_score,
        writing_score
    ]).reshape(1, -1)

    if model_type == "KNN":
        pred = model_knn.predict(data)
    else:
        pred = model_nb.predict(data)

    return "Berisiko Nilai Rendah" if pred[0] == 1 else "Tidak Berisiko Nilai Rendah"

interface = gr.Interface(
    fn=predict_risk,
    inputs=[
        gr.Radio(["KNN", "Naive Bayes"], label="Pilih Model"),
        gr.Number(label="Gender (Encoded)"),
        gr.Number(label="Race/Ethnicity (Encoded)"),
        gr.Number(label="Pendidikan Orang Tua (Encoded)"),
        gr.Number(label="Lunch (Encoded)"),
        gr.Number(label="Test Preparation Course (Encoded)"),
        gr.Number(label="Nilai Matematika"),
        gr.Number(label="Nilai Membaca"),
        gr.Number(label="Nilai Menulis"),
    ],
    outputs="text",
    title="Deteksi Dini Siswa Berisiko Mendapat Nilai Rendah",
    description="Aplikasi prediksi risiko nilai rendah siswa menggunakan model Machine Learning"
)

interface.launch()

```

Aplikasi ini menampilkan form input data siswa yang terdiri dari pilihan model (KNN atau Naive Bayes), data karakteristik siswa yang sudah di-*encode* (jenis kelamin, ras, pendidikan orang tua, jenis makan siang, dan kursus persiapan ujian), serta nilai akademik siswa (matematika, membaca, dan menulis). Setelah pengguna mengisi data dan menekan tombol Submit, aplikasi akan memproses input tersebut menggunakan model machine learning yang dipilih, lalu menampilkan hasil prediksi berupa keterangan apakah siswa berisiko mendapatkan nilai rendah atau tidak berisiko.



3. KESIMPULAN

Penerapan analisis data dan machine learning dengan pendekatan CRISP-DM terbukti mampu mendukung proses deteksi dini siswa yang berisiko mendapatkan nilai rendah menggunakan dataset *Students Performance*. Melalui tahapan pemahaman bisnis, persiapan data, pemodelan menggunakan algoritma K-Nearest Neighbor (KNN) dan Naive Bayes, serta evaluasi kinerja model, sistem yang dibangun dapat mengklasifikasikan siswa berdasarkan tingkat risikonya secara efektif. Model yang dihasilkan disimpan dalam bentuk file .pkl dan diimplementasikan menggunakan Gradio, sehingga dapat digunakan kembali tanpa proses pelatihan ulang. Implementasi sistem ini diharapkan membantu pihak sekolah atau pendidik dalam melakukan intervensi akademik lebih awal guna meningkatkan prestasi belajar siswa dan mengurangi risiko nilai rendah.