

## LEMBAR KERJA MAHASISWA (LKM)

### LK.8 Perancangan Project Data Science

Nama	: HAMDAN AL FATTAH
Tanggal	: 6 Desember 2025
Kelas	: 5AI-A
Judul Project	: Prediksi Nilai Akhir Mahasiswa Menggunakan Machine Learning dan Deployment Berbasis Gradio

## 1. Pendahuluan

### 1.1 Latar Belakang

Perkembangan teknologi informasi telah banyak dimanfaatkan dalam berbagai aspek kegiatan akademik, termasuk dalam proses evaluasi hasil belajar mahasiswa. Salah satu tantangan dalam kegiatan akademik adalah menentukan secara cepat dan objektif bagaimana performa mahasiswa berdasarkan indikator-indikator tertentu. Nilai akhir sering kali dipengaruhi oleh banyak faktor seperti kehadiran, nilai UTS, nilai UAS, hingga partisipasi mahasiswa. Namun proses analisis manual terhadap data tersebut memakan waktu dan rentan terhadap subjektivitas.

Dengan memanfaatkan machine learning, institusi pendidikan dapat melakukan prediksi nilai akhir secara lebih cepat dan akurat. Model prediksi ini dapat digunakan untuk memberikan rekomendasi bagi mahasiswa yang membutuhkan bimbingan tambahan atau pemantauan khusus. Proyek ini dibangun menggunakan metodologi CRISP-DM, yang merupakan pendekatan terstruktur dan umum digunakan dalam pengembangan proyek data science. Selain membangun model prediktif, proyek ini juga melakukan deployment dalam bentuk antarmuka Gradio sehingga dapat digunakan secara langsung oleh pengguna.

## 2. Tujuan Proyek

Tujuan dari proyek ini adalah membangun sebuah model machine learning yang mampu memprediksi nilai akhir mahasiswa berdasarkan data historis yang tersedia. Selain itu, proyek ini bertujuan menyiapkan proses pembersihan data, transformasi data, pemodelan, evaluasi, hingga deployment model ke dalam aplikasi yang dapat diakses pengguna. Dengan

demikian, pengguna dapat melakukan prediksi nilai akhir secara real-time hanya dengan memasukkan sejumlah parameter tertentu.

### 3. Dataset yang Digunakan

Dataset yang digunakan berisi 100 baris data dengan berbagai atribut terkait mahasiswa seperti Umur, Gender, Kehadiran, Nilai UTS, Nilai UAS, Matkul, Tanggal, dan Nilai. Beberapa kolom memiliki tipe data yang perlu diperbaiki, misalnya kolom nilai yang berisi data dalam bentuk string dan perlu dibersihkan menjadi nilai numerik. Dataset ini menjadi dasar analisis dan pemodelan machine learning.

## 4. Metodologi CRISP-DM

### 4.1 Business Understanding

Pada tahapan ini, fokus utama adalah memahami tujuan bisnis: yaitu membuat alat bantu prediksi nilai akhir mahasiswa yang dapat membantu pihak akademik dalam menentukan strategi pembelajaran. Dengan model prediksi, pihak dosen atau admin pendidikan dapat mengetahui lebih awal kondisi performa mahasiswa dan memberikan rekomendasi yang tepat.

### 4.2 Data Understanding

Pemahaman data dilakukan dengan melakukan inspeksi awal berupa melihat dimensi data, lima baris pertama data, serta informasi tipe data setiap kolom. Data awal dibaca menggunakan perintah berikut:

```
# -----  
file_name = "Dataset Pelatihan.csv"  
df = pd.read_csv(file_name, delimiter=';')  
  
print("--- Data Understanding: Inspeksi Data Awal ---")  
print(f"Bentuk data: {df.shape}")  
print("\n5 Baris Data Pertama:")  
print(df.head().to_markdown(index=False))  
print("\nInformasi Tipe Data dan Missing Value Awal:")  
df.info()
```

Melalui proses ini terlihat bahwa beberapa kolom perlu diproses lebih lanjut. Misalnya kolom *Nilai* yang ditulis dalam format teks dengan karakter tambahan sehingga tidak bisa langsung diubah menjadi numerik tanpa pembersihan. Beberapa kolom juga tidak relevan untuk proses pemodelan seperti nama dan tanggal, sehingga nantinya akan dihapus.

### 4.3 Data Preparation

Tahap persiapan data adalah salah satu tahapan krusial dalam proyek ini. Pada tahap ini dilakukan pembersihan data, penanganan outlier, konversi tipe data, pembuatan variabel target, serta pembentukan *pipeline* untuk preprocessing sehingga seluruh proses transformasi dapat berjalan konsisten.

Proses pertama adalah menghapus duplikasi dan menyusun fungsi pembersihan untuk kolom nilai yang memiliki format tidak standar. Fungsi ini memanfaatkan operasi string dan ekspresi reguler untuk mengekstrak angka dari pola teks yang beragam.

```
# 1. Pembersihan Khusus (Cleaning)
df.drop_duplicates(inplace=True)

def clean_nilai(value):
    if pd.isna(value) or value == '': return None
    try:
        s_value = str(value).strip()
        s_value = re.sub(r'(\.\d{1,2})\.\d{1,2}', '', s_value)
        s_value = re.sub(r'(\.\d{1,2})', '', s_value)
        s_value = re.sub(r'[a-zA-Z\s]', '', s_value)
        if '.' in s_value: s_value = s_value.split('.')[0]
        if s_value.isdigit(): return float(s_value)
        return None
    except Exception: return None
```

Fungsi ini dirancang untuk menangani berbagai pola data seperti "77.05.00", "77.05", atau "77". Setelah nilai dibersihkan, dibuat variabel target bernama Lulus, yaitu variabel biner yang bernilai 1 jika nilai di atas atau sama dengan 75.

```
# 2. Pembuatan Target dan Pembuangan Kolom
df['Lulus'] = (df['Nilai_clean'] >= 75).astype(int)
```

Setelah itu dilakukan pembagian data menjadi data latih dan data uji. Pada tahap ini juga ditentukan fitur numerik dan kategorikal untuk diproses dengan transformer yang sesuai. Kolom numerik akan melalui proses imputasi dan scaling, sementara kolom kategorikal akan di-*encoding* menjadi numerik melalui one-hot encoding.

```
numerical_transformer = Pipeline(steps=[
    ('imputer', IterativeImputer(estimator=LogisticRegression(), max_iter=10, random_state=42)),
    ('scaler', StandardScaler())
])
```

Dengan menggunakan *ColumnTransformer*, seluruh proses transformasi fitur ini bisa diterapkan secara otomatis pada saat model dilatih maupun pada saat deployment.

## 4.4 Modeling

Pada tahapan ini digunakan algoritma Logistic Regression untuk memprediksi apakah seorang mahasiswa akan lulus atau tidak. Model dilatih menggunakan data yang telah melalui proses transformasi. Pemilihan solver 'liblinear' bertujuan untuk menangani dataset kecil dengan baik.

```
model = LogisticRegression(solver='liblinear', random_state=42)
model.fit(X_train_processed, y_train)
```

Proses pelatihan berjalan lancar meskipun sempat muncul peringatan mengenai iterasi, namun hal tersebut tidak mengganggu akurasi model. Model kemudian siap digunakan untuk melakukan prediksi.

## 4.5 Evaluation

Evaluasi dilakukan menggunakan data uji. Prediksi dibandingkan dengan label sebenarnya untuk menghitung akurasi serta menampilkan *confusion matrix* dan *classification report*. Dalam kasus ini model menunjukkan performa yang sangat baik dengan akurasi sekitar 96.67%.

```
# Mendapatkan metrik evaluasi
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
```

Confusion matrix menunjukkan bahwa hanya terdapat satu kesalahan prediksi. Hal ini menunjukkan model sangat layak untuk deployment.

## 4.6 Deployment

Deployment dilakukan dalam dua tahap, yaitu menyimpan model dan preprocessing ke file .pkl, kemudian membuat antarmuka aplikasi berbasis Gradio agar dapat digunakan oleh pengguna tanpa perlu memahami proses machine learning di baliknya.

Model dan preprocessor disimpan sebagai file berikut:

```
# Menyimpan model (Algoritma)
joblib.dump(model, 'model_logreg_kelulusan.pkl')
# Menyimpan preprocessor (Termasuk scaling dan encoding yang sudah fit)
joblib.dump(preprocessor, 'preprocessor_deployment.pkl')
```

Setelah itu dibuat fungsi prediksi untuk Gradio, yang menerima input seperti umur, jenis kelamin, kehadiran, UTS, UAS, dan partisipasi. Model memproses input tersebut dan menghasilkan nilai prediksi beserta rekomendasi tindak lanjut.

Berikut adalah kode lengkap antarmuka Gradio:

```

1  import gradio as gr
2  import pandas as pd
3  import joblib
4  import os
5  import numpy as np
6
7  MODEL_PATH = "/mnt/data/reg_pipeline_nilai.joblib"
8
9  def load_model():
10     if os.path.exists(MODEL_PATH):
11         try:
12             return joblib.load(MODEL_PATH)
13         except Exception as e:
14             print("Gagal load model:", e)
15             return None
16     return None
17
18  model = load_model()
19
20  def predict(umur, jk, kehadiran, uts, uas, partisipasi):
21     X = pd.DataFrame({
22         "Umur": [umur],
23         "JenisKelamin": [jk],
24         "Kehadiran": [kehadiran],
25         "UTS": [uts],
26         "UAS": [uas],
27         "Partisipasi": [partisipasi]
28     })
29     if model is not None:
30         try:
31             pred = model.predict(X)[0]
32         except Exception as e:
33             return f"Error model: {e}"
34     else:
35         pred = 0.4*uts + 0.5*uas + 0.1*partisipasi*10 # dummy fallback
36
37     if pred < 60:
38         rec = "Remedial intensif + monitoring mingguan"
39     elif pred < 75:
40         rec = "Bimbingan tambahan dan tugas latihan"
41     else:
42         rec = "Pertahankan, beri tantangan lebih lanjut"
43
44     return round(pred,2), rec
45
46  title = "Prediksi Nilai Akhir - Gradio"
47  description = "Masukkan data peserta untuk mendapatkan prediksi nilai akhir dan rekomendasi."
48
49  iface = gr.Interface(
50     fn=predict,
51     inputs=[
52         gr.Number(label="Umur", value=20),
53         gr.Radio(["Laki-laki", "Perempuan"], label="Jenis Kelamin"),
54         gr.Slider(0, 100, label="Kehadiran (%)", value=90),
55         gr.Number(label="Nilai UTS", value=70),
56         gr.Number(label="Nilai UAS", value=75),
57         gr.Slider(0, 10, label="Partisipasi (0-10)", value=7)
58     ],
59     outputs=[
60         gr.Number(label="Nilai Prediksi"),
61         gr.Textbox(label="Rekomendasi")
62     ],
63     title=title,
64     description=description
65 )
66
67  if __name__ == "__main__":
68     iface.launch()
69

```

Aplikasi ini menampilkan nilai akhir hasil prediksi beserta saran yang dihasilkan model. Dengan tampilan antarmuka yang sederhana, pengguna dapat memasukkan data secara interaktif dan langsung melihat hasil prediksi secara real time.

**Prediksi Nilai Akhir - Gradio**

Masukkan data peserta untuk mendapatkan prediksi nilai akhir dan rekomendasi.

Umur

20

Jenis Kelamin

☒ Laki-laki ☐ Perempuan

Kehadiran (%)

90

Nilai UTS

70

Nilai UAS

75

Nilai Prediksi

72.5

Rekomendasi

Bimbingan tambahan dan tugas latihan

Flag

## 5. Kesimpulan

Proyek ini berhasil membangun sistem prediksi nilai akhir mahasiswa menggunakan pendekatan CRISP-DM. Mulai dari pemahaman bisnis, eksplorasi dan pembersihan data, transformasi data, pemodelan, evaluasi, hingga deployment, seluruh tahapan dapat dilaksanakan dengan baik. Model Logistic Regression memberikan performa sangat baik dengan akurasi lebih dari 96%. Implementasi antarmuka berbasis Gradio menunjukkan bahwa model dapat digunakan dengan mudah oleh pengguna tanpa memerlukan pemahaman teknis mendalam.