LAMPIRAN

```
import re
from turtle import onclick
import streamlit as st
import numpy as np
import pandas as pd
import joblib
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

st.set_page_config(
    page_title="Prediksi  Sleep  Disorder  Menggunakan
Metode Naive Bayes Gaussian",
    page_icon="zzz",
    layout="wide",
)
# Load model
model = joblib.load('shl_model.pkl')
if 'page' not in st.session_state:
    st.session_state.page = 'input'
def reset():
    st.session_state.page = 'input'
def handle_prediction():
    input_data = np.array([[
        st.session_state.gender,
        st.session_state.age,
        st.session_state.sleep_duration,
        st.session_state.quality_of_sleep,
        st.session_state.physical_activity_level,
        st.session_state.stress_level,
```

```
        st.session_state.bmi_category,
        st.session_state.systolic_blood_pressure,
        st.session_state.diastolic_blood_pressure,
        st.session_state.heart_rate,
        st.session_state.daily_steps
    ]])
    st.session_state.prediction                    =
model.predict(input_data)[0]
    st.session_state.page = 'result'
# handle file upload & result
def handle_file_prediction(file):
    df = pd.read_csv(file, delimiter=";")

    required_columns  =  ['Gender',  'Age',  'Sleep
Duration', 'Quality of Sleep',
                         'Physical  Activity  Level',
'Stress Level', 'BMI Category',
                         'Systolic  Blood  Pressure',
'Diatolic Blood Pressure', 'Heart Rate',
                         'Daily Steps']

    # Check all required columns
    if  not  all(col  in  df.columns  for  col  in
required_columns):
        st.error("File  tidak  memiliki  kolom  yang
diperlukan. Setidaknya file harus memiliki kolom:",
required_columns)
        return

    # Cleaning & mapping BMI Category to numerical
values
```

```
    if '1 Weight' in df['BMI Category'].values:
        df['BMI       Category']       =       df['BMI
Category'].replace('1 Weight', '1')

        # Pemetaan nilai ke integer
        df['BMI       Category']       =       df['BMI
Category'].map({'1': 1, '2': 2, '3': 3})

    nan_Sleep_Disorder           =           df['Sleep
Disorder'].isnull().any()
    if nan_Sleep_Disorder:
        df['Sleep     Disorder']     =     df['Sleep
Disorder'].fillna('None')

    # Split data features and labels (jika label 'Sleep
Disorder' ada dalam file)
    X = df[required_columns]
    y = df['Sleep Disorder'] if 'Sleep Disorder' in
df.columns else None

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

    # Train the model
    model.fit(X_train, y_train)

    # Prediksi test set
    y_pred = model.predict(X_test)

    # Accuracy model
```

```
    accuracy = accuracy_score(y_test, y_pred)

    st.write("### Hasil Prediksi dari Data File yang
Diunggah")
    st.write("**Akurasi pada Data Training:**",
model.score(X_train, y_train), " || **Akurasi pada Data
Testing:**", accuracy)

    df['predictions'] = model.predict(X)

    st.write("### Data Pelatihan (Training Data)")
    df_train = X_train.copy()
    df_train['Sleep Disorder'] = y_train
    st.dataframe(df_train)

    st.write("### Data Pengujian (Testing Data)")
    df_test = X_test.copy()
    df_test['Sleep Disorder'] = y_test
    st.dataframe(df_test)

# Input page manual data entry
if st.session_state.page == 'input':
    st.title('😴Prediksi Sleep Disorder Menggunakan
Metode Naive Bayes Gaussian😴')
    st.write("**Aplikasi ini membantu mendeteksi
potensi gangguan tidur berdasarkan data kesehatan
Anda.**")

    uploaded_file = st.file_uploader("Unggah file CSV
untuk prediksi", type=["csv"])
    if uploaded_file is not None:
```

```
        handle_file_prediction(uploaded_file)

    with st.form("input_form"):
        col1,   space1,   col2,   space2,   col3   =
st.columns([1, 0.05, 1, 0.05, 1])

        with col1:
            st.session_state.gender                  =
st.selectbox('Jenis      Kelamin',      ['Laki-laki',
'Perempuan'], index=0)
            st.session_state.gender     =     1     if
st.session_state.gender == 'Laki-laki' else 0
            st.session_state.age                     =
st.number_input('Usia', min_value=1)
            st.session_state.sleep_duration          =
st.number_input('Durasi  Tidur  (jam)',  min_value=0.0,
max_value=24.0)
            st.session_state.quality_of_sleep        =
st.number_input('Kualitas Tidur (1-10)', min_value=1,
max_value=10)

        with col2:
            st.session_state.systolic_blood_pressure
= st.number_input('Tekanan Darah Sistolik (mmHg)')
            st.session_state.diastolic_blood_pressure
= st.number_input('Tekanan Darah Diastolik (mmHg)')
            st.session_state.stress_level            =
st.number_input('Tingkat  Stres  (1-10)',  min_value=1,
max_value=10)
```

```
            st.session_state.physical_activity_level
=    st.number_input('Physical    Activity    Level',
min_value=0, max_value=200)

        with col3:
            st.session_state.bmi_category           =
st.selectbox('Kategori BMI', ['Normal', 'Overweight',
'Obesitas'], index=0)
            st.session_state.bmi_category   =   1   if
st.session_state.bmi_category  ==  'Normal'  else  2  if
st.session_state.bmi_category == 'Overweight' else 3
            st.session_state.heart_rate             =
st.number_input('Detak Jantung (bpm)')
            st.session_state.daily_steps            =
st.number_input('Langkah Harian', min_value=0)

        # submit form
        submitted = st.form_submit_button("Prediksi")
        if submitted:
            handle_prediction()

# Result manual input
elif st.session_state.page == 'result':
    st.title('😴Hasil Prediksi Sleep Disorder😴')
    prediction = st.session_state.prediction

    if prediction == 'None':
        st.success("**Anda   tidak   memiliki   gejala
gangguan tidur.**")
```

```
        st.write("Tetaplah menjaga pola hidup sehat seperti mengonsumsi makanan sehat, rajin berolahraga, dan tidur dengan teratur.")
    elif prediction == 'Insomnia':
        st.warning("**Anda memiliki gejala gangguan tidur Insomnia.**")
        st.write("""
        Berikut adalah beberapa tips mengatasi insomnia ringan:
        - Membuat lingkungan yang nyaman untuk tidur
        - Rajin berolahraga
        - Mengonsumsi makanan sehat
        - Menghindari penggunaan ponsel atau alat elektronik lainnya sebelum tidur
        - Membuat jadwal tidur yang teratur
        """)
    elif prediction == 'Sleep Apnea':
        st.error("**Anda memiliki gejala gangguan tidur Sleep Apnea.**")
        st.write("""
        Cobalah memulai pola hidup sehat seperti mengonsumsi makanan sehat, rajin berolahraga, dan tidur dengan teratur.
        Jika kondisi tidak membaik, segeralah periksa ke dokter untuk pengecekan dan penanganan lebih lanjut.
        """)

    if st.button('Prediksi Ulang'):
        reset()
```