**LAMPIRAN**

**Lampiran 1.** Kode Lampiran model Machine Learning konseptual

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error

import numpy as np

# --- Lampiran: Kode Python untuk Pemodelan Konseptual Machine Learning ---

# Kode ini mendemonstrasikan proses konseptual penggunaan model Machine Learning

# (Linear Regression) untuk memprediksi harga beras ritel berdasarkan variabel

# ekonomi, produksi, dan iklim, sesuai dengan metodologi pada karya tulis.

# Baca data yang sudah diperbaiki

df = pd.read\_csv('Data Fiks.xlsx - DATA FIKS.csv')

# 1. Pra-pemrosesan Data dan Pemilihan Fitur

# Kita akan menggunakan variabel-variabel ini untuk memprediksi harga beras ritel.

# Fitur (X): Variabel yang mempengaruhi, seperti produksi padi dan inflasi.

# Target (y): Variabel yang ingin diprediksi, yaitu harga beras ritel.

features = [

'Produksi Padi (Ton)',

'Inflasi Makanan YoY (%)',

'Total Curah Hujan (mm)',

'Akses Internet Penduduk (%)',

'Panjang Jalan Km'

]

target = 'Harga Beras (Rp/Kg) (Harga Ritel)'

# Pastikan tidak ada missing values, meskipun sudah diperiksa sebelumnya.

df\_clean = df.dropna(subset=features + [target])

X = df\_clean[features]

y = df\_clean[target]

# 2. Pembagian Data (Training dan Testing)

# Memisahkan data menjadi data latih (untuk melatih model) dan data uji (untuk menguji performa model).

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 3. Pemilihan dan Pelatihan Model

# Menggunakan model Linear Regression sebagai contoh model prediksi.

model = LinearRegression()

model.fit(X\_train, y\_train)

# 4. Prediksi dan Evaluasi

# Menggunakan model yang sudah dilatih untuk memprediksi harga pada data uji.

y\_pred = model.predict(X\_test)

# Menghitung metrik evaluasi

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

mae = mean\_absolute\_error(y\_test, y\_pred)

r2 = model.score(X\_test, y\_test)

# --- Output untuk Lampiran ---

print("Lampiran: Kode Python untuk Pemodelan Konseptual Machine Learning\n")

print("---------------------------------------------------------------")

print("Variabel Prediktor (Fitur):", features)

print("Variabel Target:", target)

print("\n--- Hasil Evaluasi Model pada Data Uji ---")

print(f"Mean Squared Error (MSE): {mse:.2f}")

print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")

print(f"Mean Absolute Error (MAE): {mae:.2f}")

print(f"R-squared (R²): {r2:.2f}")

print("\n--- Contoh Prediksi vs Nilai Aktual ---")

# Menampilkan beberapa contoh prediksi

predictions\_df = pd.DataFrame({'Nilai Aktual': y\_test, 'Prediksi': y\_pred})

print(predictions\_df.sample(5).to\_string())

# Catatan: Kode ini adalah contoh konseptual. Untuk hasil yang lebih akurat, diperlukan model yang lebih kompleks dan data yang lebih spesifik.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

**Lampiran 2.** Kode Lampiran Tren Inflasi Pangan

# --- Lampiran: Kode Python untuk Grafik Tren Inflasi Pangan ---

# Kode ini berfungsi untuk memvisualisasikan tren inflasi pangan tahunan di setiap

# provinsi di Sumatera, yang merupakan indikator kunci dari stabilitas harga dan

# ketahanan pangan di wilayah tersebut.

# Baca data yang sudah diperbaiki

df = pd.read\_csv('Data Fiks.xlsx - DATA FIKS.csv')

# Set a professional style for the plots

sns.set\_theme(style="whitegrid")

# Create a figure and axes for the plot

plt.figure(figsize=(15, 8))

# Get the unique list of provinces

provinces = df['Provinsi'].unique()

# Plot the data for each province

for province in provinces:

province\_data = df[df['Provinsi'] == province]

plt.plot(province\_data['Tahun'], province\_data['Inflasi Makanan YoY (%)'], marker='o', label=province)

# Add titles and labels

plt.title('Tren Inflasi Pangan Tahunan per Provinsi di Sumatera (2010-2025)', fontsize=16)

plt.xlabel('Tahun', fontsize=12)

plt.ylabel('Inflasi Makanan YoY (%)', fontsize=12)

# Add a legend to differentiate the provinces

plt.legend(title='Provinsi', bbox\_to\_anchor=(1.05, 1), loc='upper left')

# Adjust layout to prevent labels from overlapping and ensure readability

plt.xticks(province\_data['Tahun'], rotation=45)

plt.grid(True)

plt.tight\_layout()

# Simpan grafik

plt.savefig('inflasi\_pangan\_sumatera.png')

# Catatan: Kode ini adalah versi untuk lampiran, jadi tidak ada output grafik yang ditampilkan.

**Lampiran 3.** Kode untuk Grafik Disparitas Harga Pangan.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# --- Lampiran: Kode Python untuk Grafik Disparitas Harga Pangan ---

# Kode ini memvisualisasikan rata-rata selisih harga (gap margin) antara

# harga ritel dan harga petani untuk komoditas utama di setiap provinsi.

# Ini adalah analisis kunci untuk mengukur efisiensi rantai pasok.

# Baca data yang sudah diperbaiki

df = pd.read\_csv('Data Fiks.xlsx - DATA FIKS.csv')

# --- Data Preparation and Calculation ---

price\_columns\_retail = ['Harga Beras (Rp/Kg) (Harga Ritel)', 'Harga Cabai (Rp/Kg) (Harga Ritel)', 'Harga Bawang (Rp/Kg) (Harga Ritel)']

price\_columns\_petani = ['Harga Beras (Rp/Kg) (Harga Petani)', 'Harga Cabai (Rp/Kg) (Harga Petani)', 'Harga Bawang (Rp/Kg) (Harga Petani)']

avg\_retail\_prices = df.groupby('Provinsi')[price\_columns\_retail].mean().reset\_index()

avg\_petani\_prices = df.groupby('Provinsi')[price\_columns\_petani].mean().reset\_index()

avg\_retail\_prices.columns = ['Provinsi', 'Beras\_Ritel', 'Cabai\_Ritel', 'Bawang\_Ritel']

avg\_petani\_prices.columns = ['Provinsi', 'Beras\_Petani', 'Cabai\_Petani', 'Bawang\_Petani']

price\_disparity\_df = pd.merge(avg\_retail\_prices, avg\_petani\_prices, on='Provinsi')

melted\_df = price\_disparity\_df.melt(id\_vars='Provinsi', var\_name='Jenis\_Harga', value\_name='Harga\_Rata\_Rata')

melted\_df[['Komoditas', 'Tipe']] = melted\_df['Jenis\_Harga'].str.split('\_', expand=True)

# --- Data Visualization ---

sns.set\_theme(style="whitegrid")

plt.figure(figsize=(20, 10))

ax = sns.barplot(x='Provinsi', y='Harga\_Rata\_Rata', hue='Jenis\_Harga', data=melted\_df, palette='muted')

plt.title('Rata-rata Disparitas Harga Pangan per Komoditas dan Provinsi (2010-2025)', fontsize=16)

plt.xlabel('Provinsi', fontsize=12)

plt.ylabel('Harga Rata-rata (Rp/Kg)', fontsize=12)

plt.xticks(rotation=45, ha='right')

handles, labels = ax.get\_legend\_handles\_labels()

labels = [

'Beras (Ritel)', 'Cabai (Ritel)', 'Bawang (Ritel)',

'Beras (Petani)', 'Cabai (Petani)', 'Bawang (Petani)'

]

ax.legend(handles, labels, title='Harga', bbox\_to\_anchor=(1.05, 1), loc='upper left')

for p in ax.patches:

ax.annotate(f'{p.get\_height():,.0f}', (p.get\_x() + p.get\_width() / 2., p.get\_height()),

ha='center', va='center', fontsize=9, color='black', xytext=(0, 5),

textcoords='offset points')

plt.tight\_layout()

plt.savefig('disparitas\_harga\_sumatera.png')

# Catatan: Kode ini adalah versi untuk lampiran, jadi tidak ada output grafik yang ditampilkan.

print("Kode Python untuk Grafik Disparitas Harga Pangan sudah siap.")

**Lampiran 4.** Kode Untuk Dampak Iklim Pada Produksi

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# --- Lampiran: Kode Python untuk Grafik Dampak Iklim pada Produksi ---

# Kode ini berfungsi untuk mengukur dan memvisualisasikan dampak perubahan curah

# hujan terhadap produksi padi dengan menghitung koefisien korelasi Pearson.

# Ini adalah analisis kuantitatif untuk memahami hubungan iklim-produksi.

# Baca data yang sudah diperbaiki

df = pd.read\_csv('Data Fiks.xlsx - DATA FIKS.csv')

# --- Data Preparation and Calculation ---

correlation = df.groupby('Provinsi').apply(lambda x: x['Produksi Padi (Ton)'].corr(x['Total Curah Hujan (mm)']))

correlation = correlation.to\_frame(name='Koefisien\_Korelasi').reset\_index()

correlation = correlation.sort\_values(by='Koefisien\_Korelasi', ascending=False)

# --- Presentation ---

sns.set\_theme(style="whitegrid")

plt.figure(figsize=(10, 7))

sns.barplot(x='Provinsi', y='Koefisien\_Korelasi', data=correlation, palette='coolwarm')

plt.title('Dampak Perubahan Curah Hujan terhadap Produksi Padi (Koefisien Korelasi)', fontsize=16)

plt.xlabel('Provinsi', fontsize=12)

plt.ylabel('Koefisien Korelasi', fontsize=12)

plt.xticks(rotation=45, ha='right')

for index, row in correlation.iterrows():

plt.text(correlation.index.get\_loc(index), row['Koefisien\_Korelasi'], f"{row['Koefisien\_Korelasi']:.2f}", ha='center', va='bottom', fontsize=12)

plt.tight\_layout()

plt.savefig('korelasi\_produksi\_curah\_hujan.png')

# Catatan: Kode ini adalah versi untuk lampiran, jadi tidak ada output grafik yang ditampilkan.

print("Kode Python untuk Grafik Dampak Iklim pada Produksi sudah siap.")