

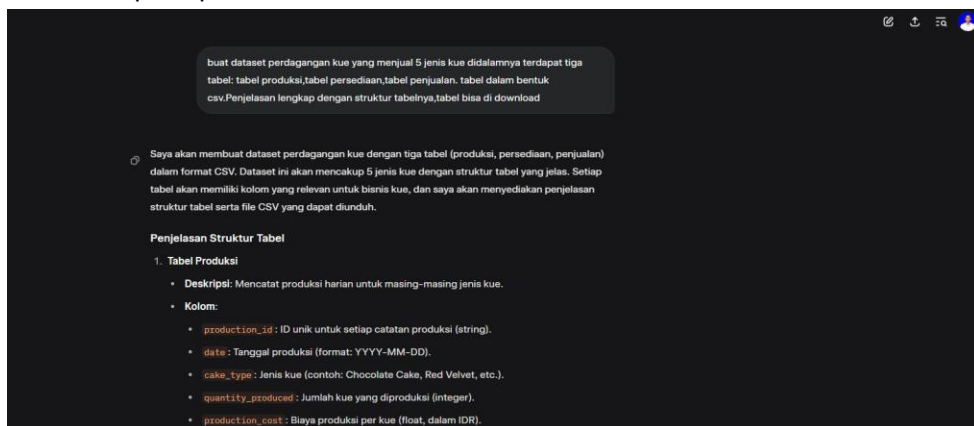
Nama : Beneto Apsa Hizkia Tambunan
Nim : 12030123120018
Kelas : Pengkodean dan Pemrograman / F

PERTEMUAN 13

Phyton

Pada pertemuan kali ini mahasiswa mempelajari dan menjalankan dasar-dasar pemrograman Python melalui W3Schools dan Google Colab. Tujuan dari tugas ini adalah untuk membangun pemahaman konsep dasar Python serta memberikan pengalaman dalam menggunakan kode Python, misalnya untuk membuat analisis data sederhana terkait perdagangan. Mahasiswa juga diarahkan untuk menampilkan hasil analisis dalam bentuk tabel dan visualisasi data secara langsung di platform Google Colab.

1. masukkan prompt ke Grok AI



2. masukkan ketiga file tersebut ke Google Colab, kemudian execute

Gemini X Files

sales.csv inventory.csv production.csv

Happy to help! Here's a set of tasks I can execute in your notebook to help answer your question.

- **Data loading:** Load the "sales.csv", "inventory.csv", and "production.csv" datasets into dataframes.
- **Data exploration:** Explore each dataset to understand their shape, columns, data types, and basic statistics.
- **Data analysis:** Analyze each dataset independently to identify key patterns, distributions, and potential relationships within each file.
- **Data cleaning:** Perform basic data cleaning on each dataframe, such as handling missing values and checking for duplicates.
- **Data preparation:** Prepare the dataframes for potential merging or further analysis by ensuring consistent column names and data types where relevant.
- **Finish task:** Summarize the findings from the data exploration and analysis of each dataset, providing an overview of their contents and potential insights they offer.

Want anything changed? Feel free to send feedback, and I will try to update the plan accordingly!

Execute plan

Rate this answer

Executed plan:

- ✓ Preparing to execute plan
- ✓ Data loading
- ✓ Data exploration
- ✓ Data analysis
- ✓ Data cleaning
- ✓ Data preparation
- ✓ Finish task

I've finished executing the plan for you!

3. Hasil dari Google Colab

Subtask:

Load the "sales.csv", "inventory.csv", and "production.csv" datasets into dataframes.

Reasoning: Load the three CSV files into pandas DataFrames as instructed.

```
import pandas as pd

df_sales = pd.read_csv('sales.csv')
df_inventory = pd.read_csv('inventory.csv')
df_production = pd.read_csv('production.csv')
```

Reasoning: Display the head of each loaded DataFrame to verify the loading process.

```
[2] display(df_sales.head())
display(df_inventory.head())
display(df_production.head())
```

	sale_id	date	cake_type	quantity_sold	sale_price	total_revenue
0	SALE001	2025-06-01	Chocolate Cake	20	25000.0	500000.0
1	SALE002	2025-06-01	Red Velvet	15	30000.0	450000.0
2	SALE003	2025-06-01	Cheese Cake	10	35000.0	350000.0
3	SALE004	2025-06-01	Black Forest	12	28000.0	336000.0
4	SALE005	2025-06-01	Vanilla Sponge	25	20000.0	500000.0

	inventory_id	date	cake_type	stock_quantity
0	INV001	2025-06-01	Chocolate Cake	50

▼ Data exploration

Subtask:

Explore each dataset to understand their shape, columns, data types, and basic statistics.

Reasoning: Explore the shape, columns, data types, and basic statistics of each dataframe as requested in the instructions.

```
print("--- df_sales ---")
print("Shape:", df_sales.shape)
print("\nInfo:")
df_sales.info()
print("\nDescribe:")
display(df_sales.describe())

print("\n--- df_inventory ---")
print("Shape:", df_inventory.shape)
print("\nInfo:")
df_inventory.info()
print("\nDescribe:")
display(df_inventory.describe())

print("\n--- df_production ---")
print("Shape:", df_production.shape)
print("\nInfo:")
df_production.info()
print("\nDescribe:")
display(df_production.describe())
```

▼ Data analysis

Subtask:

Analyze each dataset independently to identify key patterns, distributions, and potential relationships within each file.

Reasoning: Convert the 'date' columns in all dataframes to datetime objects to enable time series analysis.

```
df_sales['date'] = pd.to_datetime(df_sales['date'])
df_inventory['date'] = pd.to_datetime(df_inventory['date'])
df_production['date'] = pd.to_datetime(df_production['date'])

print("Date columns converted to datetime objects.")
```

↻ Date columns converted to datetime objects.

Reasoning: Analyze the distributions and relationships in `df_sales` as per instruction 1 and examine unique cake types across all dataframes as per instruction 4.

```
[5] import matplotlib.pyplot as plt
import seaborn as sns

# Instruction 1: Analyze df_sales
print("--- Analyzing df_sales ---")
plt.figure(figsize=(18, 5))

plt.subplot(1, 3, 1)
sns.histplot(df_sales['quantity_sold'], kde=True)
```

dataframes as per instruction 5.

```
# Instruction 2: Analyze df_inventory
print("--- Analyzing df_inventory ---")
plt.figure(figsize=(8, 5))
sns.histplot(df_inventory['stock_quantity'], kde=True)
plt.title('Distribution of Stock Quantity')
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(x='cake_type', y='stock_quantity', data=df_inventory)
plt.title('Stock Quantity vs Cake Type')
plt.show()

plt.figure(figsize=(12, 6))
sns.lineplot(x='date', y='stock_quantity', hue='cake_type', data=df_inventory)
plt.title('Stock Quantity over Time by Cake Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Instruction 5: Check for trends over time using the date column
print("\n--- Time Trends (Sales) ---")
plt.figure(figsize=(12, 6))
sns.lineplot(x='date', y='total_revenue', hue='cake_type', data=df_sales)
plt.title('Total Revenue over Time by Cake Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

print("\n--- Time Trends (Production) ---")
plt.figure(figsize=(12, 6))
sns.lineplot(x='date', y='quantity_produced', hue='cake_type', data=df_production)
plt.title('Quantity Produced over Time by Cake Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

