

**Nama : Feby Wahyu Pristitasari**  
**NIM : 12030123140306**  
**Kelas : F – PENGKODEAN DAN PEMROGRAMAN**

**PERTEMUAN 11-12**  
**DATASET PERDAGANGAN – PYTHON GOOGLE COLLAB**

*Prompt: “Buat dataset perdagangan yang didalamnya terdapat tiga tabel: tabel produksi, tabel persediaan, tabel penjualan, tabel dalam bentuk csv, penjelasan lengkap dengan struktur tabelnya.”*

**Penjelasan Struktur Tabel**

**1. Tabel Produksi**

Berisi data tentang jumlah barang yang diproduksi oleh suatu perusahaan pada periode tertentu.

Kolom:

- **production\_id** : ID unik untuk setiap catatan produksi (integer, primary key).
- **product\_id** : ID produk yang diproduksi (integer, foreign key jika ada).
- **product\_name** : Nama produk (string).
- **production\_date** : Tanggal produksi (date, format YYYY-MM-DD).
- **quantity\_produced** : Jumlah unit yang diproduksi (integer).
- **production\_cost** : Biaya produksi per unit dalam Rupiah (decimal).

Production_id	Product_id	Product_name	Production_date	Quantity_produced	Production_cost
1	101	Beras Premium	45672	1000	12000.5
2	101	Beras Premium	45698	800	12250.75
3	102	Minyak Goreng	45677	500	15000
4	102	Minyak Goreng	45703	600	14800.25
5	103	Gula Pasir	45682	1200	13000
6	103	Gula Pasir	45708	1000	13250.5

**2. Tabel Persediaan**

Berisi data tentang stok barang yang tersedia di gudang pada waktu tertentu.

Kolom:

- **inventory\_id** : ID unik untuk setiap catatan persediaan (integer, primary key).
- **product\_id** : ID produk yang ada di persediaan (integer, foreign key jika ada).
- **product\_name** : Nama produk (string).
- **inventory\_date** : Tanggal pencatatan persediaan (date, format YYYY-MM-DD).
- **quantity\_in\_stock** : Jumlah unit yang tersedia di stok (integer).
- **warehouse\_location** : Lokasi gudang penyimpanan (string).

Inventory_id	Product_id	Product_name	Inventory_date	Quantity_in_stock	Warehouse_location
1	101	Beras Premium	45688	1200	Jakarta
2	101	Beras Premium	45716	1500	Jakarta
3	102	Minyak Goreng	45688	700	Surabaya
4	102	Minyak Goreng	45716	900	Surabaya
5	103	Gula Pasir	45688	1000	Bandung
6	103	Gula Pasir	45716	800	Bandung

3. Tabel Penjualan

Berisi data tentang penjualan produk pada periode tertentu.

Kolom:

- sale\_id : ID unik untuk setiap catatan penjualan (integer, primary key).
- product\_id : ID produk yang terjual (integer, foreign key ke tabel produk jika ada).
- product\_name : Nama produk (string).
- sale\_date : Tanggal penjualan (date, format YYYY-MM-DD).
- quantity\_sold : Jumlah unit yang terjual (integer).
- sale\_price : Harga jual per unit dalam Rupiah (decimal).
- total\_revenue : Total pendapatan dari penjualan (quantity\_sold \* sale\_price, decimal).

Sale_id	Product_id	Product_name	Sale_date	Quantity_sold	Sale_price	Total_revenue
1	101	Beras Premium	45677	300	15000	4500000
2	101	Beras Premium	45703	400	15250	6100000
3	102	Minyak Goreng	45682	200	18000	3600000
4	102	Minyak Goreng	45708	250	18250	4562500
5	103	Gula Pasir	45687	500	16000	8000000
6	103	Gula Pasir	45713	400	16250	6500000

Prompt: “Berdasarkan dataset perdagangan yang berisi ketiga tabel di bawah ini, buat data analitik sederhana lengkap dengan download hasil tabel dan visualisasi hasilnya.”

--- sertakan file csv ke Gemini dalam Google Collab, kemudian jalankan task ---

File Edit Lihat Sisipkan Runtime Fitur Bantuan

Perintah Kode Teks Jalankan semua

Task

Berdasarkan dataset perdagangan yang berisi ketiga tabel di bawah ini, buat data analitik sederhana lengkap dengan download hasil tabel dan visualisasi hasilnya

Here is all the data you need: "sales.csv" "inventory.csv" "production.csv"

Data loading

Subtask:

Load the provided CSV files into pandas DataFrames.

Reasoning: Load the three CSV files into pandas DataFrames and display the first 5 rows of each to verify.

```
[1] import pandas as pd

sales_df = pd.read_csv('sales.csv')
inventory_df = pd.read_csv('inventory.csv')
production_df = pd.read_csv('production.csv')

display(sales_df.head())
display(inventory_df.head())
display(production_df.head())
```

	sale_id	product_id	product_name	sale_date	quantity_sold	sale_price	total_revenue
0	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0
1	2	101	Beras Premium	2025-02-15	400	15250.0	
2	3	102	Minyak Goreng	2025-01-25	200	18000.0	

15.40 Python 3

Gemini X

Berdasarkan dataset perdagangan yang berisi ketiga tabel di bawah ini, buat data analitik sederhana lengkap dengan download hasil tabel dan visualisasi hasilnya

sales.csv inventory.csv production.csv

Senang bisa membantu. Berikut serangkaian tugas yang bisa saya eksekusi di notebook Anda untuk membantu menjawab pertanyaan Anda.

- Data loading:** Load the three datasets ('sales.csv', 'inventory.csv', and 'production.csv') into separate pandas DataFrames.
- Data exploration:** Explore each DataFrame individually to understand its structure, data types, and identify any missing values or inconsistencies. Determine the key columns and their relationships across the three datasets.
- Data preparation:** Prepare the data for merging and analysis. This may involve handling missing values (e.g., imputation or removal), converting data types if necessary, and creating new features if needed (e.g., calculated fields based on existing columns like total sales value).
- Data wrangling:** Merge the three datasets based on common keys (e.g., product ID, date, etc.) into a single DataFrame. This will create a comprehensive dataset for analysis. Clean the merged data, removing any duplicate entries and handling any remaining inconsistencies.
- Data analysis:** Perform descriptive statistics on the combined dataset. Calculate key metrics such as total sales, average inventory levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.
- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter

Masukkan perintah di sini

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

File Edit Lihat Sisipkan Runtime Fitur Bantuan

Perintah Kode Teks Jalankan semua

Data loading

Subtask:

Load the provided CSV files into pandas DataFrames.

Reasoning: Load the three CSV files into pandas DataFrames and display the first 5 rows of each to verify.

```
import pandas as pd

sales_df = pd.read_csv('sales.csv')
inventory_df = pd.read_csv('inventory.csv')
production_df = pd.read_csv('production.csv')

display(sales_df.head())
display(inventory_df.head())
display(production_df.head())
```

	sale_id	product_id	product_name	sale_date	quantity_sold	sale_price	total_revenue
0	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0
1	2	101	Beras Premium	2025-02-15	400	15250.0	6100000.0
2	3	102	Minyak Goreng	2025-01-25	200	18000.0	3600000.0
3	4	102	Minyak Goreng	2025-02-20	250	18250.0	4562500.0
4	5	103	Gula Pasir	2025-01-30	500	16000.0	8000000.0

	inventory_id	product_id	product_name	inventory_date	quantity_in_stock	warehouse_location
0	1	101	Beras Premium	2025-01-31	1200	Jakarta
1	2	101	Beras Premium	2025-02-28	1500	Jakarta
2	3	102	Minyak Goreng	2025-01-31	700	Surabaya
3	4	102	Minyak Goreng	2025-02-28	900	Surabaya
4	5	103	Gula Pasir	2025-01-31	1000	Bandung

	production_id	product_id	product_name	production_date	quantity_produced	production_cost
0	1	101	Beras Premium	2025-01-15	1000	12000.50
1	2	101	Beras Premium	2025-02-10	800	12250.75
2	3	102	Minyak Goreng	2025-01-20	500	15000.00

15.49 Python 3

Gemini X NameError

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

Rencana yang dieksekusi:

- Bersiap mengeksekusi rencana
- Data loading
- Data exploration
- Data preparation
- Data wrangling
- Data analysis
- Data visualization
- Data preparation
- Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

File Edit Lihat Sisipkan Runtime Fitur Bantuan

Perintah Kode Teks Jalankan semua

Data exploration

Subtask:

Explore the loaded datasets to understand their structure, identify missing values, and determine key relationships between them.

Reasoning: Examine the first few rows, data types, descriptive statistics, and shape of each DataFrame; identify missing values; analyze key columns and potential relationships between DataFrames; and check for duplicate rows.

```
# Data Exploration and Cleaning

# 1. Basic DataFrame Information
print("Sales DataFrame Information:")
display(sales_df.head())
print(sales_df.dtypes)
display(sales_df.describe(include='all'))
print(sales_df.shape)

print("\nInventory DataFrame Information:")
display(inventory_df.head())
print(inventory_df.dtypes)
display(inventory_df.describe(include='all'))
print(inventory_df.shape)

print("\nProduction DataFrame Information:")
display(production_df.head())
print(production_df.dtypes)
display(production_df.describe(include='all'))
print(production_df.shape)

# 2. Missing Values
print("\nMissing Values:")
print("Sales DataFrame:\n", sales_df.isnull().sum())
print("Inventory DataFrame:\n", inventory_df.isnull().sum())
print("Production DataFrame:\n", production_df.isnull().sum())

# 3. Key Relationships and Potential Merging Columns
print("\nKey Relationships and Merging Columns:")
print("The 'product_id' column is a key column common across all three datasets and can be used for merging.")
print("The 'sale_date', 'inventory_date', and 'production_date' columns represent time series data and can be used to analyze trends and relationships over time.")
print("The 'quantity_sold', 'quantity_in_stock', and 'quantity_produced' columns represent quantities and can be analyzed to understand the relationships between")

# 4. Duplicate Rows
print("\nDuplicate Rows:")
```

15.49 Python 3

Gemini X NameError

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

Rencana yang dieksekusi:

- Bersiap mengeksekusi rencana
- Data loading
- Data exploration
- Data preparation
- Data wrangling
- Data analysis
- Data visualization
- Data preparation
- Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

File

Edit

Lihat

Sleipkan

Runtime

Fitur

Bantuan

Perintah

+ Kode

+ Teks

Jalankan semua

Data preparation

Subtask:

Prepare the data for merging and analysis. Handle potential missing values, convert data types if necessary, and create new features.

Reasoning: Handle missing values, convert data types, and engineer new features as per the instructions.

```
[21] # Missing Value Handling (No missing values found in initial exploration, but rechecking)
for df in [sales_df, inventory_df, production_df]:
    for col in ['quantity_sold', 'quantity_in_stock', 'quantity_produced']:
        if col in df.columns and df[col].isnull().any():
            df[col].fillna(df[col].median(), inplace=True)

# Data Type Conversion
for df in [sales_df, inventory_df, production_df]:
    for col in ['sale_date', 'inventory_date', 'production_date']:
        if col in df.columns:
            try:
                df[col] = pd.to_datetime(df[col])
            except Exception as e:
                print(f"Error converting {col} in {df}: {e}")

# Feature Engineering
sales_df['total_sale_value'] = sales_df['quantity_sold'] * sales_df['sale_price']

# Data Consistency Checks (Example: check for negative quantities)
for df in [sales_df, inventory_df, production_df]:
    for col in ['quantity_sold', 'quantity_in_stock', 'quantity_produced']:
        if col in df.columns:
            if df[col] < 0:
                print(f"Warning: Negative values found in {col} of {df}")
                # Handle negative values (e.g., replace with 0 or absolute value)
                df[col] = df[col].abs()

display(sales_df.head())
display(inventory_df.head())
display(production_df.head())
```

	sale_id	product_id	product_name	sale_date	quantity_sold	sale_price	total_revenue	total_sale_value
0	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0	4500000.0
1	2	101	Beras Premium	2025-02-15	400	15250.0	6100000.0	6100000.0

Gemini

X

NameError

...

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

Rencana yang dieksekusi:

- Berisap mengeksekusi rencana
- Data loading
- Data exploration
- Data preparation
- Data wrangling
- Data analysis
- Data visualization
- Data preparation
- Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

0/2000

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan code dengan hati-hati. [Pelajari lebih lanjut](#)

File

Edit

Lihat

Sleipkan

Runtime

Fitur

Bantuan

Perintah

+ Kode

+ Teks

Jalankan semua

Data wrangling

Subtask:

Merge the three datasets: sales\_df, inventory\_df, and production\_df into a single DataFrame called merged\_df.

Reasoning: Merge sales and inventory dataframes, then merge the result with production dataframe. Remove duplicates and display the result.

```
[22] # Merge sales and inventory DataFrames
sales_inventory_df = pd.merge(sales_df, inventory_df, on=['product_id', 'product_name'], how='inner', suffixes=('_sales', '_inventory'))

# Merge the result with production DataFrame
merged_df = pd.merge(sales_inventory_df, production_df, on=['product_id', 'product_name'], how='inner', suffixes=('_', '_sales_inventory', '_production'))

# Remove duplicate rows
merged_df.drop_duplicates(inplace=True)

# Display the merged DataFrame
print(merged_df.shape)
display(merged_df.head())
```

	sale_id	product_id	product_name	sale_date	quantity_sold	sale_price	total_revenue	total_sale_value	inventory_id	inventory_date	quantity_in_stock
0	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0	4500000.0	1	2025-01-31	1200
1	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0	4500000.0	1	2025-01-31	1200
2	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0	4500000.0	2	2025-02-28	1500
3	1	101	Beras Premium	2025-01-20	300	15000.0	4500000.0	4500000.0	2	2025-02-28	1500
4	2	101	Beras Premium	2025-02-15	400	15250.0	6100000.0	6100000.0	1	2025-01-31	1200

Gemini

X

NameError

...

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

Rencana yang dieksekusi:

- Berisap mengeksekusi rencana
- Data loading
- Data exploration
- Data preparation
- Data wrangling
- Data analysis
- Data visualization
- Data preparation
- Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

0/2000

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan code dengan hati-hati. [Pelajari lebih lanjut](#)

File

Edit

Lihat

Sleipkan

Runtime

Fitur

Bantuan

Perintah

+ Kode

+ Teks

Jalankan semua

Data analysis

Subtask:

Analyze the merged dataset to identify key trends and relationships.

Reasoning: Calculate descriptive statistics for numerical columns in merged\_df and analyze trends over time by grouping data by date and calculating aggregate metrics. Visualize these trends using line plots.

```
import matplotlib.pyplot as plt

# Descriptive statistics
numerical_cols = ['quantity_sold', 'sale_price', 'total_revenue', 'total_sale_value', 'quantity_in_stock', 'quantity_produced', 'production_cost']
display(merged_df[numerical_cols].describe())

# Trends over time
merged_df['sale_date'] = pd.to_datetime(merged_df['sale_date'])
merged_df['inventory_date'] = pd.to_datetime(merged_df['inventory_date'])
merged_df['production_date'] = pd.to_datetime(merged_df['production_date'])
time_series_df = merged_df.groupby('sale_date').agg({'total_revenue': 'sum', 'quantity_produced': 'sum', 'quantity_in_stock': 'mean'})

# Plotting
plt.figure(figsize=(12, 6))
plt.plot(time_series_df.index, time_series_df['total_revenue'], label='Total Revenue', color='blue')
plt.plot(time_series_df.index, time_series_df['quantity_produced'], label='Total Quantity Produced', color='green')
plt.plot(time_series_df.index, time_series_df['quantity_in_stock'], label='Average Quantity in Stock', color='red')

plt.xlabel('Sale Date')
plt.ylabel('Value')
plt.title('Trends Over Time')
plt.legend()
plt.grid(True)
plt.show()
```

	quantity_sold	sale_price	total_revenue	total_sale_value	quantity_in_stock	quantity_produced	production_cost
count	24.000000	24.000000	2.400000e+01	2.400000e+01	24.000000	24.000000	24.000000
mean	344.666667	16458.333333	5.543750e+06	5.543750e+06	1016.666667	850.000000	13383.666667
std	100.909102	1290.426786	1.508666e+06	1.508666e+06	272.933607	248.473601	1178.121235
min	200.000000	15000.000000	3.600000e+06	3.600000e+06	700.000000	500.000000	12000.500000
25%	250.000000	15250.000000	4.500000e+06	4.500000e+06	800.000000	600.000000	12250.750000

Gemini

X

NameError

...

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

Rencana yang dieksekusi:

- Berisap mengeksekusi rencana
- Data loading
- Data exploration
- Data preparation
- Data wrangling
- Data analysis
- Data visualization
- Data preparation
- Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

0/2000

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan code dengan hati-hati. [Pelajari lebih lanjut](#)

File Edit Lihat Sisipkan Runtime Filur Bantuan

Perintah Kode Teks Jalankan semua

quantity\_sold sale\_price total\_revenue total\_sale\_value quantity\_in\_stock quantity\_produced production\_cost

count	24.000000	24.000000	2.400000e+01	2.400000e+01	24.000000	24.000000	24.000000
mean	341.666667	16458.333333	5.543750e+06	5.543750e+06	1016.666667	800.000000	13383.666667
std	103.909102	1280.426786	1.508666e+06	1.508666e+06	272.933607	248.473601	1178.121235
min	200.000000	15000.000000	3.600000e+06	3.600000e+06	700.000000	500.000000	12000.500000
25%	250.000000	15250.000000	4.500000e+06	4.500000e+06	800.000000	600.000000	12250.750000
50%	350.000000	16125.000000	5.331250e+06	5.331250e+06	900.000000	900.000000	13125.250000
75%	400.000000	18000.000000	6.500000e+06	6.500000e+06	1200.000000	1000.000000	14800.250000
max	500.000000	18250.000000	8.000000e+06	8.000000e+06	1500.000000	1200.000000	15000.000000

Trends Over Time

Reasoning: Explore potential correlations between sales, inventory, and production by calculating correlation coefficients and creating scatter

infer to visualize relationships

Terminal

15.49

15.49

15.49

File Edit Lihat Sisipkan Runtime Filur Bantuan

Perintah Kode Teks Jalankan semua

```
plt.subplot(1, 2, 2)
plt.scatter(merged_df['quantity_sold'], merged_df['quantity_produced'], color='green')
plt.xlabel('Quantity Sold')
plt.ylabel('Quantity Produced')
plt.title('Quantity Sold vs. Quantity Produced')

plt.subplot(1, 2, 3)
plt.scatter(merged_df['quantity_in_stock'], merged_df['quantity_produced'], color='red')
plt.xlabel('Quantity in Stock')
plt.ylabel('Quantity Produced')
plt.title('Quantity in Stock vs. Quantity Produced')

plt.tight_layout()
plt.show()
```

quantity\_sold quantity\_in\_stock quantity\_produced

quantity_sold	1.000000	0.189078	0.841993
quantity_in_stock	0.189078	1.000000	0.285268
quantity_produced	0.841993	0.285268	1.000000

Quantity Sold vs. Quantity in Stock

Quantity Sold vs. Quantity Produced

Quantity in Stock vs. Quantity Produced

Langkah berikutnya: [buat kode dengan correlation\\_matrix](#) [Lihat plot yang direkomendasikan](#) [New Interactive sheet](#)

15.49

15.49

15.49

File Edit Lihat Sisipkan Runtime Filur Bantuan

Perintah Kode Teks Jalankan semua

```
plt.xlabel('Product ID')
plt.ylabel('Average Inventory')
plt.title('Average Inventory per Product')

plt.subplot(1, 2, 3)
plt.bar(product_performance_index, product_performance['total_production_cost'], color='lightcoral')
plt.xlabel('Product ID')
plt.ylabel('Total Production Cost')
plt.title('Total Production Cost per Product')

plt.tight_layout()
plt.show()
```

Total Sales per Product

Average Inventory per Product

Total Production Cost per Product

Reasoning: Investigate sales seasonality by analyzing sales patterns across different months or quarters and visualize seasonal trends using a line plot.

Terminal

15.49

15.49

15.49

File Edit Lihat Simpan Runtime Filur Bantuan

Perintah + Kode + Teks Jalankan semua

# Sales seasonality analysis

merged\_df['sale\_month'] = merged\_df['sale\_date'].dt.month

monthly\_sales = merged\_df.groupby('sale\_month')['total\_revenue'].sum()

# Plotting monthly sales

plt.figure(figsize=(10, 6))

plt.plot(monthly\_sales.index, monthly\_sales.values, marker='o', linestyle='--')

plt.xlabel('Month')

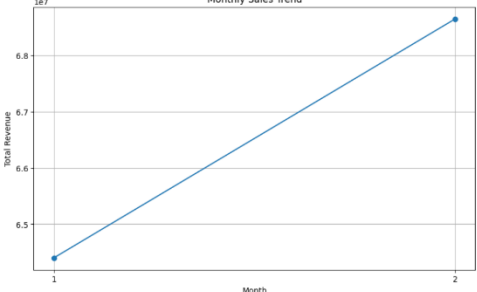
plt.ylabel('Total Revenue')

plt.title('Monthly Sales Trend')

plt.xticks(monthly\_sales.index) # Ensure all month labels are displayed

plt.grid(True)

plt.show()



Month	Total Revenue
1	6.45
2	6.85

**Reasoning:** Examine production efficiency by analyzing the relationship between production costs and quantities produced. Create a scatter plot to show cost vs. quantity and identify potential patterns.

Generasi X NameError

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

- Rencana yang dieksekusi:
  - Bersiap mengeksekusi rencana
  - Data loading
  - Data exploration
  - Data preparation
  - Data wrangling
  - Data analysis
  - Data visualization
  - Data preparation
  - Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

0/2000

Generasi dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

15.49

File Edit Lihat Simpan Runtime Filur Bantuan

Perintah + Kode + Teks Jalankan semua

# Production efficiency analysis

plt.figure(figsize=(8, 6))

plt.scatter(merged\_df['quantity\_produced'], merged\_df['production\_cost'], color='purple')

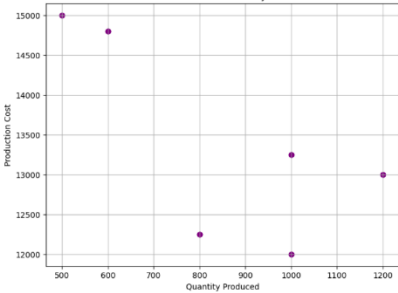
plt.xlabel('Quantity Produced')

plt.ylabel('Production Cost')

plt.title('Production Cost vs. Quantity Produced')

plt.grid(True)

plt.show()



Quantity Produced	Production Cost
500	15000
600	14800
800	12200
1000	13200
1200	13000

**Reasoning:** Examine production efficiency by analyzing the relationship between production costs and quantities produced. Create a scatter plot to show cost vs. quantity and identify potential patterns.

Generasi X NameError

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

- Rencana yang dieksekusi:
  - Bersiap mengeksekusi rencana
  - Data loading
  - Data exploration
  - Data preparation
  - Data wrangling
  - Data analysis
  - Data visualization
  - Data preparation
  - Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

0/2000

Generasi dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

15.49

File Edit Lihat Simpan Runtime Filur Bantuan

Perintah + Kode + Teks Jalankan semua

# Enhanced time-series plot

plt.figure(figsize=(14, 7))

plt.plot(time\_series\_df.index, time\_series\_df['total\_revenue'], label='Total Revenue', color='blue', marker='o')

plt.plot(time\_series\_df.index, time\_series\_df['quantity\_produced'], label='Quantity Produced', color='green', marker='x')

plt.plot(time\_series\_df.index, time\_series\_df['quantity\_in\_stock'], label='Average Quantity in Stock', color='red', marker='x')

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

plt.ylabel('Value', fontsize=12)

plt.title('Time Series Analysis of Revenue, Production, and Inventory', fontsize=14)

plt.legend(fontsize=10)

plt.grid(True, linestyle='--', alpha=0.7)

plt.xticks(rotation=45)

plt.tight\_layout()

plt.savefig('time\_series\_plot.png')

plt.show()

# Enhanced correlation analysis scatter plots

plt.figure(figsize=(15, 5))

plt.subplot(1, 2, 1)

plt.scatter(merged\_df['quantity\_sold'], merged\_df['quantity\_in\_stock'], color='blue', label='Data Points')

plt.xlabel('Quantity Sold', fontsize=10)

plt.ylabel('Quantity in Stock', fontsize=10)

plt.title('Quantity Sold vs. Quantity in Stock', fontsize=12)

plt.legend()

plt.subplot(1, 2, 2)


plt.scatter(merged\_df['quantity\_sold'], merged\_df['quantity\_produced'], color='green', label='Data Points')

plt.xlabel('Quantity Sold', fontsize=10)

plt.ylabel('Quantity Produced', fontsize=10)

plt.title('Quantity Sold vs. Quantity Produced', fontsize=12)

plt.legend()



Quantity Sold	Quantity in Stock	Quantity Produced
500	15000	15000
600	14800	14800
800	12200	12200
1000	13200	13200
1200	13000	13000

**Reasoning:** Enhance the visualizations generated in the previous step by adding titles, axis labels, legends, and improving the overall aesthetics. Save the plots as image files.

Generasi X NameError

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Finish task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

- Rencana yang dieksekusi:
  - Bersiap mengeksekusi rencana
  - Data loading
  - Data exploration
  - Data preparation
  - Data wrangling
  - Data analysis
  - Data visualization
  - Data preparation
  - Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

0/2000

Generasi dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

15.49

p13.ipynb

File Edit Lihat Sisipan Runtime Fitur Bantuan

Perintah Kode Teks Jalankan semua

**Data preparation**

Subtask:

Prepare the final analytical table for download.

Reasoning: Select relevant columns from `merged_df`, rename them for clarity, sort by `sale_date`, and save the resulting analytical table as a CSV file.

```

# Select relevant columns
analytical_df = merged_df[['sale_date', 'product_id', 'product_name', 'quantity_sold', 'total_revenue', 'quantity_in_stock', 'quantity_produced', 'production_cost']]

# Rename columns for clarity
analytical_df = analytical_df.rename(columns={
    'sale_date': 'Sale Date',
    'product_id': 'Product ID',
    'product_name': 'Product Name',
    'quantity_sold': 'Quantity Sold',
    'total_revenue': 'Total Revenue',
    'quantity_in_stock': 'Quantity in Stock',
    'quantity_produced': 'Quantity Produced',
    'production_cost': 'Production Cost'
})

# Sort by sale date
analytical_df = analytical_df.sort_values(by='Sale Date')

# Save the analytical table as a CSV file
analytical_df.to_csv('analytical_table.csv', index=False)

display(analytical_df.head())

```

	Sale Date	Product ID	Product Name	Quantity Sold	Total Revenue	Quantity in Stock	Quantity Produced	Production Cost
0	2025-01-20	101	beras premium	300	4500000.0	1200	1000	12000.50
1	2025-01-20	101	beras premium	300	4500000.0	1200	800	12250.75
2	2025-01-20	101	beras premium	300	4500000.0	1500	1000	12000.50
3	2025-01-20	101	beras premium	300	4500000.0	1500	800	12250.75
10	2025-01-25	102	minyak goreng	200	3600000.0	900	500	15000.00

Gemini X NameError

levels, production costs, and profitability. Analyze trends over time, and identify any correlations between sales, inventory, and production. Explore potential insights related to product performance, sales seasonality, and production efficiency.

- Data visualization:** Create visualizations to represent the key findings. This could include line charts for sales trends over time, bar charts for product performance comparison, scatter plots to show correlations, and other relevant chart types.
- Data preparation:** Prepare the final analytical table for download. This may involve selecting specific columns, sorting data, and formatting the table for easy readability.
- Final task:** Generate a concise report summarizing the analysis, including key findings, insights, and visualizations. Include the downloadable analytical table along with the visualizations.

Mau mengubah sesuatu? Silakan kirim masukan dan saya akan mencoba menyesuaikan rencananya.

Jalankan rencana

Berikan penilaian untuk jawaban ini

Rencana yang dieksekusi:

- Bersiap mengeksekusi rencana
- Data loading
- Data exploration
- Data preparation
- Data wrangling
- Data analysis
- Data visualization
- Data preparation
- Finish task

Saya telah mengeksekusi rencana untuk Anda.

Berikan penilaian untuk jawaban ini

Masukkan perintah di sini

Gemini dapat membuat kesalahan, jadi periksa kembali responsnya dan gunakan kode dengan hati-hati. [Pelajari lebih lanjut](#)

0/2000

✓ 15.49

kembali ke Grok untuk meminta membuat kode python sebelum di upload kembali ke Google Collab dengan Gemini

berdasarkan 3 tabel yang telah dibuatkan, buatlah analisis data dan juga visualisasi hasilnya yang bisa langsung di download

Saya akan membuat analisis data berdasarkan tiga tabel (Produksi, Persediaan, Penjualan) yang telah dibuat sebelumnya, lalu menyediakan visualisasi dalam bentuk grafik menggunakan Python dengan pustaka `pandas` dan `matplotlib`. Visualisasi akan mencakup tren produksi, persediaan, dan penjualan, serta hubungan antar variabel seperti total pendapatan. File Python akan menghasilkan grafik yang dapat diunduh sebagai file PNG. Selain itu, saya akan menyertakan analisis dalam format teks dan kode visualisasi dalam tag `<aiArtifact>`.

**Analisis Data**

- Tren Produksi:**
  - Tabel produksi menunjukkan bahwa Beras Premium diproduksi sebanyak 1000 unit pada Januari 2025 dan 800 unit pada Februari 2025, menunjukkan penurunan produksi sebesar 20%.
  - Minyak Goreng meningkat dari 500 unit (Januari) menjadi 600 unit (Februari), naik 20%.
  - Gula Pasir menurun dari 1200 unit (Januari) menjadi 1000 unit (Februari), turun sekitar 16.67%.
  - Biaya produksi per unit cenderung meningkat sedikit dari Januari ke Februari untuk semua produk (misalnya, Beras Premium dari 12,000.50 ke 12,250.75 IDR).
- Tren Persediaan:**
  - Berdasarkan Beras Premium meningkat dari 1500 unit (Januari) menjadi 1600

Bagaimana Grok bisa membantu?

DeepSearch Think

Grok 3

<> Kode ▶ Pratinjau

Grafik telah disimpan sebagai 'trade\_analysis\_visualization.png'

The first chart is a grouped bar chart titled 'Perbandingan Produk, Persediaan, dan Penjualan (Jan-Feb 2025)'. It compares three metrics (Produksi, Persediaan, Penjualan) for three products (Beras Premium, Minyak Goreng, Gula Pasir) across January and February 2025. The second chart is a line graph titled 'Tren Total Pendapatan per Produk (Jan-Feb 2025)'. It shows the total revenue trend for the same three products over the same period.



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import io
import base64

# Data dari tabel CSV (dibaca langsung dari string untuk simulasi)
production_data = """
production_id,product_id,product_name,production_date,quantity_produced,production_cost
1,101,Beras Premium,2025-01-15,1000,12000.50
2,101,Beras Premium,2025-02-10,800,12250.75
3,102,Minyak Goreng,2025-01-20,500,15000.00
4,102,Minyak Goreng,2025-02-15,600,14800.25
5,103,Gula Pasir,2025-01-25,1200,13000.00
6,103,Gula Pasir,2025-02-20,1000,13250.50
"""

inventory_data = """
inventory_id,product_id,product_name,inventory_date,quantity_in_stock,warehouse_location
1,101,Beras Premium,2025-01-31,1200,Jakarta
2,101,Beras Premium,2025-02-28,1500,Jakarta
3,102,Minyak Goreng,2025-01-31,700,Surabaya
4,102,Minyak Goreng,2025-02-28,900,Surabaya
5,103,Gula Pasir,2025-01-31,1000,Bandung
6,103,Gula Pasir,2025-02-28,800,Bandung
"""

sales_data = """
sale_id,product_id,product_name,sale_date,quantity_sold,sale_price,total_revenue
1,101,Beras Premium,2025-01-20,300,15000.00,4500000.00
2,101,Beras Premium,2025-02-15,400,15250.00,6100000.00
3,102,Minyak Goreng,2025-01-25,200,10000.00,3600000.00
4,102,Minyak Goreng,2025-02-20,250,10250.00,4562500.00
5,103,Gula Pasir,2025-01-30,500,16000.00,8000000.00
6,103,Gula Pasir,2025-02-25,400,16250.00,6500000.00
"""

# Membaca data dari string ke DataFrame
production_df = pd.read_csv(io.StringIO(production_data))
inventory_df = pd.read_csv(io.StringIO(inventory_data))
sales_df = pd.read_csv(io.StringIO(sales_data))

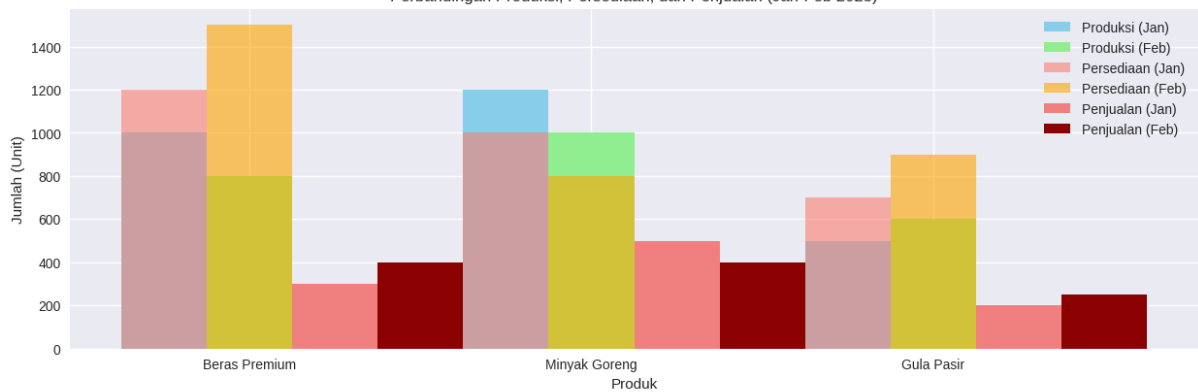
# Mengubah kolom tanggal menjadi datetime dan mengekstrak bulan
production_df['month'] = pd.to_datetime(production_df['production_date']).dt.strftime('%b')
inventory_df['month'] = pd.to_datetime(inventory_df['inventory_date']).dt.strftime('%b')
sales_df['month'] = pd.to_datetime(sales_df['sale_date']).dt.strftime('%b')

# Mengatur gaya visualisasi
plt.style.use('seaborn-v0.8')

# Enhanced sales seasonality plot
plt.figure(figsize=(10, 6))
plt.plot(monthly_sales.index, monthly_sales.values, marker='o')
plt.xlabel('Month', fontsize=12)
plt.ylabel('Total Revenue', fontsize=12)
plt.title('Monthly Sales Seasonality', fontsize=14)
plt.xticks(monthly_sales.index)
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.savefig('sales_seasonality.png')
plt.show()

# Enhanced production efficiency plot
plt.figure(figsize=(8, 6))
plt.scatter(merged_df['quantity_produced'], merged_df['production_cost'], s=100)
z = np.polyfit(merged_df['quantity_produced'], merged_df['production_cost'], 2)
p = np.poly1d(z)
plt.plot(merged_df['quantity_produced'], p(merged_df['quantity_produced']), color='red', linewidth=2)
plt.xlabel('Quantity Produced', fontsize=12)
plt.ylabel('Production Cost', fontsize=12)
plt.title('Production Efficiency: Cost vs. Quantity', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.tight_layout()
plt.savefig('production_efficiency.png')
plt.show()
```

Perbandingan Produksi, Persediaan, dan Penjualan (Jan-Feb 2025)



Tren Total Pendapatan per Produk (Jan-Feb 2025)

