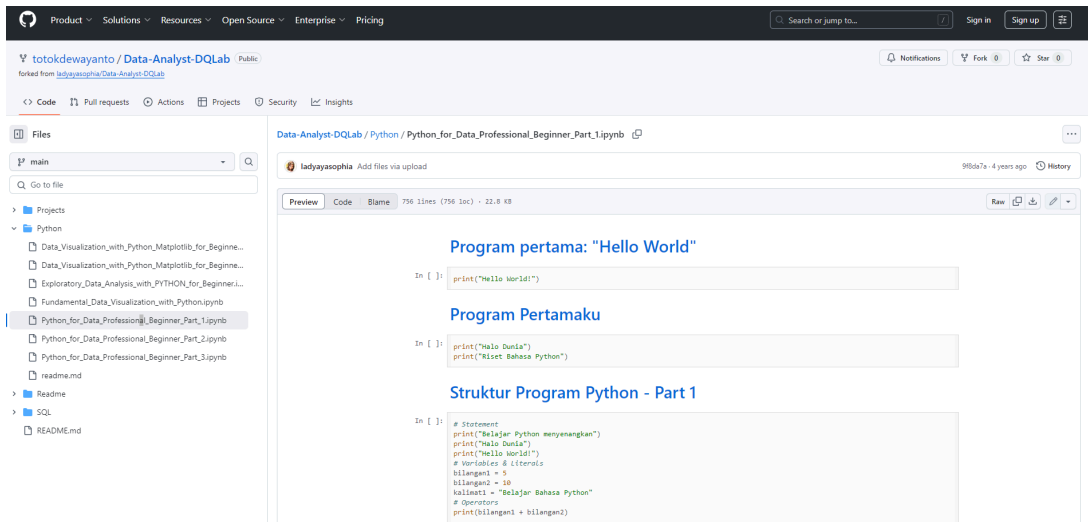


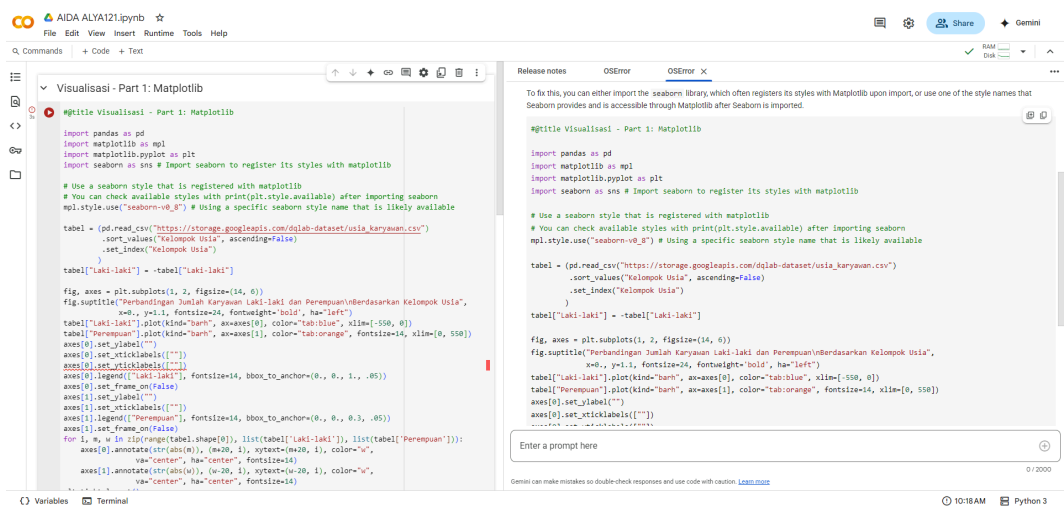
Nama : Aida Alya Rahmadani
NIM : 12030123130121
Kelas : E
Mata Kuliah : Pengkodean dan Pemrograman
Dosen Pengampu : Dr. Totok Dewayanto, S.E., M.Si., Akt.

PERTEMUAN KE-12
PYTHON

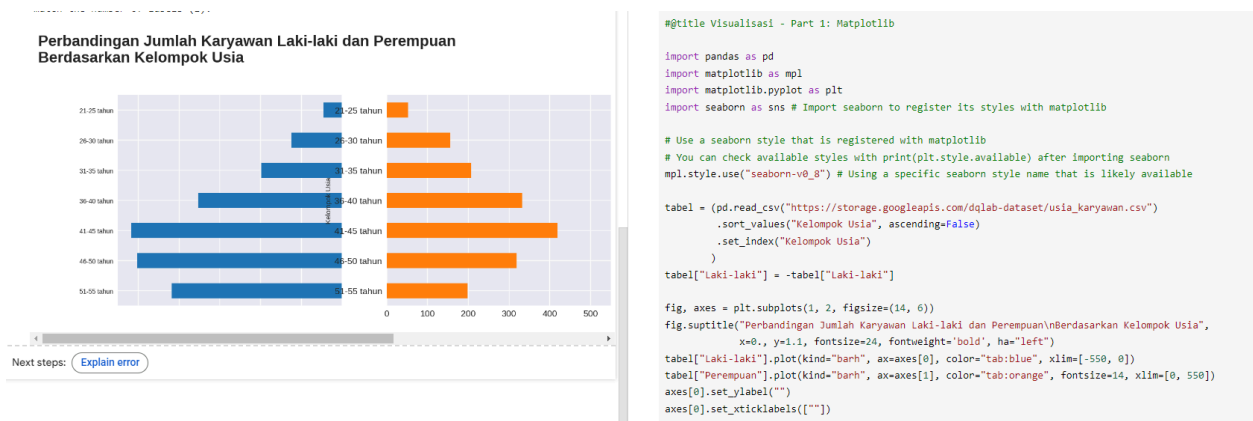
Referensi



Google Colab

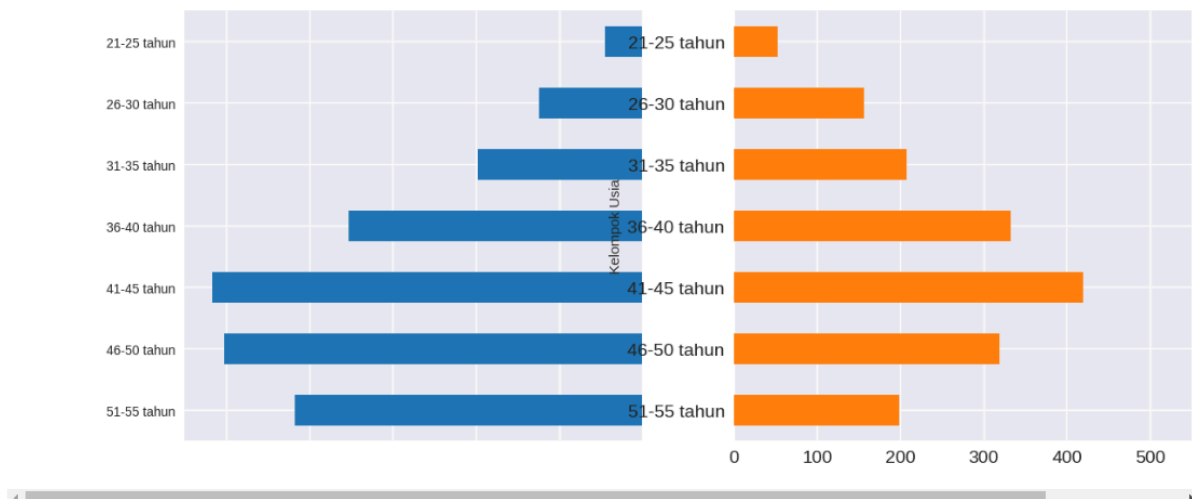


Output dan Kode Baru



Visualisasi Hasil

Perbandingan Jumlah Karyawan Laki-laki dan Perempuan Berdasarkan Kelompok Usia



DATASET PERDAGANGAN

Perintah: Buatkan dataset dengan nama “perdagangan” dengan 3 tabel yaitu tabel produksi, tabel persediaan, dan tabel penjualan dalam bentuk CSV. Buat lengkap dengan struktur tabelnya!

Grok

buatkan dataset dengan nama perdagangan dengan 3 tabel yaitu tabel produksi, tabel persediaan, dan tabel penjualan. Tolong lengkap dengan struktur tabelnya

Saya akan membuat dataset untuk sistem perdagangan dengan 3 tabel: Produksi, Persediaan, dan Penjualan, lengkap dengan struktur tabelnya. Tabel akan dirancang dalam format SQL untuk memudahkan implementasi ke dalam database relasional. Setiap tabel akan memiliki kolom yang relevan untuk konteks perdagangan, dengan primary key dan foreign key untuk menjaga relasi antar tabel.

Dataset Perdagangan SQL

sql

Show inline

Penjelasan Struktur Tabel:

1. Tabel Produksi:

- `id_produk`: Primary key, auto-increment untuk identifikasi unik.
- `kode_barang`: Kode unik untuk barang, digunakan sebagai referensi di tabel lain.
- `nama_barang`: Nama barang yang diproduksi.
- `tanggal_produk`: Tanggal produksi barang.
- `jumlah_produk`: Jumlah barang yang diproduksi.
- `satuan`: Satuan ukuran barang (misal: Pack, Box).
- `biaya_produk`: Biaya produksi total.
- `created_at`: Waktu pembuatan record.

How can Grok help?

DeepSearch

Think

Grok 3

Grok

Berikut adalah dataset untuk sistem perdagangan dengan tiga tabel (Produksi, Persediaan, dan Penjualan) dalam format CSV. Setiap tabel disertakan dengan struktur kolom dan contoh data yang sesuai dengan struktur yang telah dijelaskan sebelumnya.

Tabel Produksi CSV

csv

Show inline

Tabel Persediaan CSV

csv

Show inline

Tabel Penjualan CSV

csv

Show inline

Penjelasan:

- Setiap file CSV mencerminkan struktur tabel yang sama seperti pada versi SQL sebelumnya.
- Kolom `created_at` menggunakan waktu saat ini (2025-05-22 10:27:00) sesuai informasi yang diberikan.
- Data contoh sama dengan yang digunakan pada versi SQL untuk konsistensi.
- Anda dapat mengunduh file CSV di atas atau menyalin isinya untuk digunakan dalam aplikasi seperti spreadsheet atau database yang mendukung impor CSV.
- Relasi antar tabel tetap dipertahankan melalui kolom `kode_barang`, meskipun dalam format CSV, integritas relasi harus dikelola secara manual saat mengimpor ke database.

Perintah: Dari ketiga tabel csv di atas buat data analitik sederhana!

dari ketiga tabel csv di atas buat data analitik sederhana

Untuk membuat analitik sederhana dari tiga tabel CSV (Produksi, Persediaan, dan Penjualan), saya akan menganalisis data untuk menghasilkan wawasan yang bermanfaat, seperti total produksi, rata-rata stok, tingkat penjualan per barang, dan margin keuntungan sederhana. Analisis akan dilakukan berdasarkan data yang telah diberikan (masing-masing tabel memiliki 10 baris data). Saya akan menyajikan hasil analitik dalam bentuk teks dan tabel ringkas, serta menyertakan visualisasi berupa grafik untuk memperjelas temuan.

Analitik Sederhana

1. Total Produksi dan Biaya Produksi

- Tujuan:** Menghitung total jumlah produksi dan total biaya produksi dari tabel Produksi.
- Perhitungan:**
 - Total jumlah produksi: Jumlahkan kolom `jumlah_produk`.
 - Total biaya produksi: Jumlahkan kolom `biaya_produk`.
- Hasil:**
 - Total produksi: 10,200 unit (dari 10 barang).
 - Total biaya produksi: Rp 39,250,000.00.

2. Rata-rata Stok per Barang

- Tujuan:** Menghitung rata-rata jumlah stok di gudang dari tabel Persediaan.

```
{
  "type": "baz",
  "data": {
    "labels": ["Kopi Bubuk", "Teh Celup", "Gula Pasir", "Minyak Goreng", "Beras Premium", "Susu Bubuk",
    "Tepung Terigu", "Mie Instan", "Sarden Kaleng", "Sabun Cair"],
    "datasets": [
      {
        "label": "Jumlah Terjual",
        "data": [50, 20, 100, 50, 20, 30, 50, 100, 20, 10],
        "backgroundColor": ["#FF6384", "#36A2EB", "#FFCE56", "#48CB9C", "#9966FF", "#FF9F40", "#C9C9C9",
        "#E7E9ED", "#76D7C4", "#F1948A"],
        "borderColor": ["#FF6384", "#36A2EB", "#FFCE56", "#48CB9C", "#9966FF", "#FF9F40", "#C9C9C9",
        "#E7E9ED", "#76D7C4", "#F1948A"],
        "borderWidth": 1
      }
    ]
  },
  "options": {
    "scales": {
      "y": {
        "beginAtZero": true,
        "title": {
          "display": true,
          "text": "Jumlah Terjual (Unit)"
        }
      },
      "x": {
        "title": {
          "display": true,
          "text": "Nama Barang"
        }
      }
    },
    "plugins": {
      "legend": {
        "display": true,
        "position": "top"
      },
      "title": {
        "display": true
      }
    }
  }
}
```

Perintah: Saya akan menggunakan python google colab untuk menyelesaikan data analitik di atas, buat tutorialnya secara urut!

Aida Alya 2.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

Files

Analyze your files with code written by Gemini Upload

sample_data

README.md

anscombe.json

california_housing_test.csv

california_housing_train.csv

mnist_test.csv

mnist_train_small.csv

penjualan.csv

persediaan.csv

produksi.csv

1/1 [1] import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files

2/1 [2] # Mengunggah file CSV
print("Silakan unggah file: produksi.csv, persediaan.csv, penjualan.csv")
uploaded = files.upload()

Memastikan semua file diunggah
expected_files = ["produksi.csv", "persediaan.csv", "penjualan.csv"]
if not all(file in uploaded for file in expected_files):
 print("Error: Pastikan Anda mengunggah ketiga file CSV.")
else:
 print("Semua file berhasil diunggah!")

Silakan unggah file: produksi.csv, persediaan.csv, penjualan.csv
3 files
• penjualan.csv(text/csv) - 938 bytes, last modified: 5/22/2025 - 100% done
• persediaan.csv(text/csv) - 841 bytes, last modified: 5/22/2025 - 100% done
• produksi.csv(text/csv) - 874 bytes, last modified: 5/22/2025 - 100% done
Saving penjualan.csv to penjualan.csv.csv
Saving persediaan.csv to persediaan.csv.csv
Saving produksi.csv to produksi.csv.csv
Error: Pastikan Anda mengunggah ketiga file CSV.

4/1 [4] # Membaca file CSV
try:
 produksi = pd.read_csv('produksi.csv')
 persediaan = pd.read_csv('persediaan.csv')
 penjualan = pd.read_csv('penjualan.csv')

 # Menampilkan nama kolom dan beberapa baris untuk verifikasi
 print("Kolom di Tabel Produksi:", list(produksi.columns))
 print("Kolom di Tabel Persediaan:", list(persediaan.columns))
 print("Kolom di Tabel Penjualan:", list(penjualan.columns))
 print("\n5 Baris Pertama Tabel Produksi:")
 print(produksi.head(5))

Tutorial: Analisis Data Perdagangan di Google Colab

Langkah 1: Persiapan di Google Colab

1. Buka Google Colab:

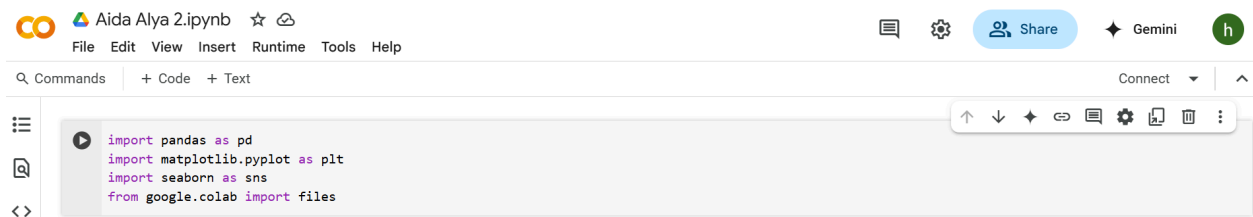
- Kunjungi [Google Colab](#) dan buat notebook baru.
- Beri nama notebook, misalnya Analisis_Perdagangan.

2. Impor Library yang Dibutuhkan:

- Kita akan menggunakan pandas untuk pengolahan data, matplotlib dan seaborn untuk visualisasi.
- Tambahkan kode berikut di sel pertama:

Kode Python

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
```



Langkah 2: Mengunggah File CSV

1. Simpan File CSV:

- Salin isi tiga file CSV dari respons sebelumnya (Produksi, Persediaan, Penjualan) ke dalam file terpisah:
 - produksi.csv
 - persediaan.csv
 - penjualan.csv
- Simpan file-file ini di komputer Anda.

2. Unggah File ke Google Colab:

- Di Google Colab, tambahkan sel kode untuk mengunggah file:

Kode Python

```
from google.colab import files
uploaded = files.upload()
```

- Klik "Choose Files" dan unggah ketiga file CSV (produksi.csv, persediaan.csv, penjualan.csv).

3. Baca File CSV ke DataFrame:

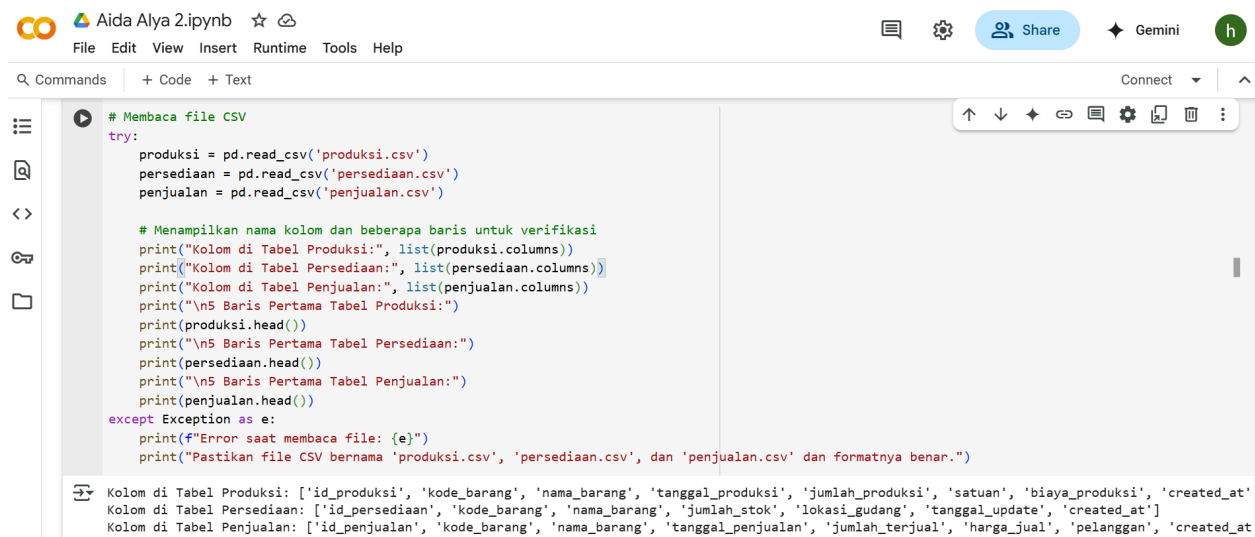
- Gunakan pandas untuk membaca file CSV ke dalam DataFrame:

Kode Python

```
# Membaca file CSV
try:
    produksi = pd.read_csv('produksi.csv')
    persediaan = pd.read_csv('persediaan.csv')
    penjualan = pd.read_csv('penjualan.csv')

    # Menampilkan nama kolom dan beberapa baris untuk verifikasi
    print("Kolom di Tabel Produksi:", list(produksi.columns))
    print("Kolom di Tabel Persediaan:", list(persediaan.columns))
    print("Kolom di Tabel Penjualan:", list(penjualan.columns))
    print("\n5 Baris Pertama Tabel Produksi:")
    print(produksi.head())
    print("\n5 Baris Pertama Tabel Persediaan:")
    print(persediaan.head())
    print("\n5 Baris Pertama Tabel Penjualan:")
    print(penjualan.head())
except Exception as e:
    print(f"Error saat membaca file: {e}")
    print("Pastikan file CSV bernama 'produksi.csv', 'persediaan.csv', dan 'penjualan.csv' dan formatnya benar.")
```

- Kode ini akan menampilkan 5 baris pertama dari setiap tabel untuk memastikan data terbaca dengan benar.



The screenshot shows a Jupyter Notebook interface with the following components:

- Top Bar:** Includes the Google Colab logo, the filename "Aida Alya 2.ipynb", and icons for file operations, settings, and sharing.
- Command Bar:** Shows "Q Commands" and options to add code or text.
- Code Editor:** Contains the Python code from the previous block, which reads three CSV files and prints their column names and the first 5 rows.
- Output:** The execution of the code has produced the following output:

```
Kolom di Tabel Produksi: ['id_produk', 'kode_barang', 'nama_barang', 'tanggal_produk', 'jumlah_produk', 'satuan', 'biaya_produk', 'created_at']
Kolom di Tabel Persediaan: ['id_persediaan', 'kode_barang', 'nama_barang', 'jumlah_stok', 'lokasi_gudang', 'tanggal_update', 'created_at']
Kolom di Tabel Penjualan: ['id_penjualan', 'kode_barang', 'nama_barang', 'tanggal_penjualan', 'jumlah_terjual', 'harga_jual', 'pelanggan', 'created_at']
```

5 Baris Pertama Tabel Produksi:				
id_produk	kode_barang	nama_barang	tanggal_produk	
0	1	BRG001 Kopi Bubuk 500g	2025-05-01	
1	2	BRG002 Teh Celup 50s	2025-05-02	
2	3	BRG003 Gula Pasir 1kg	2025-05-03	
3	4	BRG004 Minyak Goreng 1L	2025-05-04	
4	5	BRG005 Beras Premium 5kg	2025-05-05	

jumlah_produk	satuan	biaya_produk	created_at
0	1000	Pack 5000000.0	2025-05-22 10:29:00
1	800	Box 3200000.0	2025-05-22 10:29:00
2	1500	Pack 7500000.0	2025-05-22 10:29:00
3	1200	Botol 6000000.0	2025-05-22 10:29:00
4	500	Sak 2500000.0	2025-05-22 10:29:00

5 Baris Pertama Tabel Persediaan:				
id_persediaan	kode_barang	nama_barang	jumlah_stok	lokasi_gudang
0	1	BRG001 Kopi Bubuk 500g	950	Gudang Utama
1	2	BRG002 Teh Celup 50s	780	Gudang Sekunder
2	3	BRG003 Gula Pasir 1kg	1400	Gudang Utama
3	4	BRG004 Minyak Goreng 1L	1150	Gudang Utama
4	5	BRG005 Beras Premium 5kg	480	Gudang Sekunder

tanggal_update	created_at
0	2025-05-10 2025-05-22 10:29:00
1	2025-05-10 2025-05-22 10:29:00
2	2025-05-10 2025-05-22 10:29:00
3	2025-05-11 2025-05-22 10:29:00
4	2025-05-11 2025-05-22 10:29:00

5 Baris Pertama Tabel Penjualan:			
id_penjualan	kode_barang	nama_barang	tanggal_penjualan
0	1	BRG001 Kopi Bubuk 500g	2025-05-15
1	2	BRG002 Teh Celup 50s	2025-05-15
2	3	BRG003 Gula Pasir 1kg	2025-05-16
3	4	BRG004 Minyak Goreng 1L	2025-05-16
4	5	BRG005 Beras Premium 5kg	2025-05-17

jumlah_terjual	harga_jual	pelanggan	created_at
0	50	15000.0 Toko Sejahtera	2025-05-22 10:29:00
1	20	25000.0 Cafe Nusantara	2025-05-22 10:29:00
2	100	12000.0 Supermarket Maju	2025-05-22 10:29:00
3	50	20000.0 Toko Makmur	2025-05-22 10:29:00
4	20	60000.0 Warung Mapan	2025-05-22 10:29:00

Langkah 3: Analisis Data

Kita akan menghitung analitik sederhana seperti yang dilakukan sebelumnya: total produksi, rata-rata stok, tingkat penjualan, dan margin keuntungan.

1. Total Produksi dan Biaya Produksi:

- Hitung total jumlah produksi dan biaya produksi dari tabel Produksi.

Kode Python

Total produksi dan biaya produksi

```
total produksi = produksi['jumlah produksi'].sum()
```

```
total biaya produksi = produksi["biaya produksi"].sum()
```

```
print(f"Total Jumlah Produksi: {total_produksi} unit")
```

```
print(f"Total Biaya Produksi: Rp {total_biaya_produksi:,.2f}")
```

```
# Membaca file CSV
try:
    produksi = pd.read_csv('produksi.csv')
    persediaan = pd.read_csv('persediaan.csv')
    penjualan = pd.read_csv('penjualan.csv')

    # Menampilkan nama kolom dan beberapa baris untuk verifikasi
    print("Kolom di Tabel Produksi:", list(produksi.columns))
    print("Kolom di Tabel Persediaan:", list(persediaan.columns))
    print("Kolom di Tabel Penjualan:", list(penjualan.columns))
    print("\n5 Baris Pertama Tabel Produksi:")
    print(produksi.head())
    print("\n5 Baris Pertama Tabel Persediaan:")
    print(persediaan.head())
    print("\n5 Baris Pertama Tabel Penjualan:")
    print(penjualan.head())
except Exception as e:
    print(f"Error saat membaca file: {e}")
    print("Pastikan file CSV bernama 'produksi.csv', 'persediaan.csv', dan 'penjualan.csv' dan formatnya benar.")
```

Output

Kolom di Tabel Produksi: ['id_produk', 'kode_barang', 'nama_barang', 'tanggal_produk', 'jumlah_produk', 'satuan', 'biaya_produk', 'created_at']
 Kolom di Tabel Persediaan: ['id_persediaan', 'kode_barang', 'nama_barang', 'jumlah_stok', 'lokasi_gudang', 'tanggal_update', 'created_at']
 Kolom di Tabel Penjualan: ['id_penjualan', 'kode_barang', 'nama_barang', 'tanggal_penjualan', 'jumlah_terjual', 'harga_jual', 'pelanggan', 'created_at']

5 Baris Pertama Tabel Produksi:

	id_produk	kode_barang	nama_barang	tanggal_produk
0	1	BRG001	Kopi Bubuk 500g	2025-05-01
1	2	BRG002	Teh Celup 50s	2025-05-02
2	3	BRG003	Gula Pasir 1kg	2025-05-03
3	4	BRG004	Minyak Goreng 1L	2025-05-04
4	5	BRG005	Beras Premium 5kg	2025-05-05

	jumlah_produk	satuan	biaya_produk	created_at
0	1000	Pack	5000000.0	2025-05-22 10:29:00
1	800	Box	3200000.0	2025-05-22 10:29:00
2	1500	Pack	7500000.0	2025-05-22 10:29:00
3	1200	Botol	6000000.0	2025-05-22 10:29:00
4	500	Sak	2500000.0	2025-05-22 10:29:00

5 Baris Pertama Tabel Persediaan:

	id_persediaan	kode_barang	nama_barang	jumlah_stok	lokasi_gudang
0	1	BRG001	Kopi Bubuk 500g	950	Gudang Utama
1	2	BRG002	Teh Celup 50s	780	Gudang Sekunder
2	3	BRG003	Gula Pasir 1kg	1400	Gudang Utama
3	4	BRG004	Minyak Goreng 1L	1150	Gudang Utama
4	5	BRG005	Beras Premium 5kg	480	Gudang Sekunder

	tanggal_update	created_at
0	2025-05-10	2025-05-22 10:29:00
1	2025-05-10	2025-05-22 10:29:00
2	2025-05-10	2025-05-22 10:29:00
3	2025-05-11	2025-05-22 10:29:00
4	2025-05-11	2025-05-22 10:29:00

5 Baris Pertama Tabel Penjualan:

	id_penjualan	kode_barang	nama_barang	tanggal_penjualan
0	1	BRG001	Kopi Bubuk 500g	2025-05-15
1	2	BRG002	Teh Celup 50s	2025-05-15
2	3	BRG003	Gula Pasir 1kg	2025-05-16
3	4	BRG004	Minyak Goreng 1L	2025-05-16
4	5	BRG005	Beras Premium 5kg	2025-05-17

5 Baris Pertama Tabel Penjualan:

	id_penjualan	kode_barang	nama_barang	tanggal_penjualan
0	1	BRG001	Kopi Bubuk 500g	2025-05-15
1	2	BRG002	Teh Celup 50s	2025-05-15
2	3	BRG003	Gula Pasir 1kg	2025-05-16
3	4	BRG004	Minyak Goreng 1L	2025-05-16
4	5	BRG005	Beras Premium 5kg	2025-05-17

	jumlah_terjual	harga_jual	pelanggan	created_at
0	50	15000.0	Toko Sejahtera	2025-05-22 10:29:00
1	20	25000.0	Cafe Nusantara	2025-05-22 10:29:00
2	100	12000.0	Supermarket Maju	2025-05-22 10:29:00
3	50	20000.0	Toko Makmur	2025-05-22 10:29:00
4	20	60000.0	Warung Mapan	2025-05-22 10:29:00

2. Rata-rata Stok per Barang:

- Hitung rata-rata stok dari tabel Persediaan.

Kode Python

Rata-rata stok

```
total_stok = persediaan['jumlah_stok'].sum()
```

```
rata_rata_stok = total_stok / len(persediaan)
```

```
print(f"Total Stok: {total_stok} unit")
```

```
print(f"Rata-rata Stok per Barang: {rata_rata_stok:.2f} unit")
```

```
[ ] try:
    # Memastikan kolom numerik
    persediaan['jumlah_stok'] = pd.to_numeric(persediaan['jumlah_stok'], errors='coerce')

    # Menghitung rata-rata stok
    total_stok = persediaan['jumlah_stok'].sum()
    rata_rata_stok = total_stok / len(persediaan)

    print(f"Total Stok: {total_stok:,} unit")
    print(f"Rata-rata Stok per Barang: {rata_rata_stok:.2f} unit")
except KeyError as e:
    print(f"Error: Kolom {e} tidak ditemukan di tabel Persediaan.")
except Exception as e:
    print(f"Error: {e}")
```

Output

```
➤ Total Stok: 9,850 unit
  Rata-rata Stok per Barang: 985.00 unit
```

3. Tingkat Penjualan per Barang:

- Hitung total unit terjual, pendapatan, serta identifikasi barang dengan penjualan tertinggi dan terendah.

Kode Python

Total penjualan dan pendapatan

```
penjualan['pendapatan'] = penjualan['jumlah_terjual'] * penjualan['harga_jual']
```

```
total_terjual = penjualan['jumlah_terjual'].sum()
```

```
total_pendapatan = penjualan['pendapatan'].sum()
```

Barang dengan penjualan tertinggi dan terendah

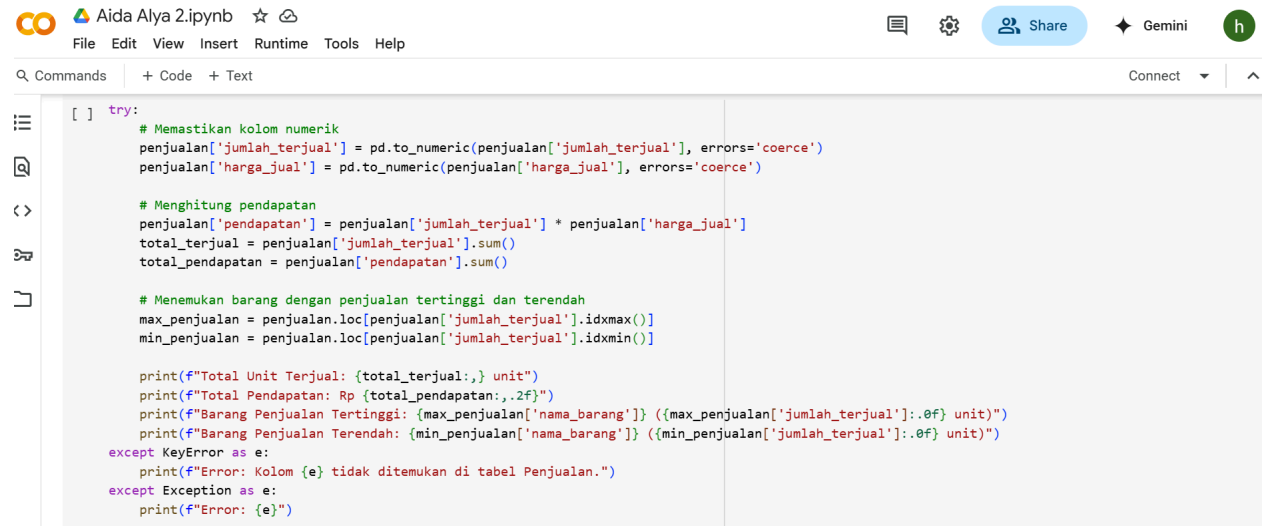
```
max_penjualan = penjualan.loc[penjualan['jumlah_terjual'].idxmax()]
```

```
min_penjualan = penjualan.loc[penjualan['jumlah_terjual'].idxmin()]
```

```

print(f'Total Unit Terjual: {total_terjual} unit")
print(f'Total Pendapatan: Rp {total_pendapatan:,.2f}")
print(f'Barang Penjualan Tertinggi: {max_penjualan['nama_barang']}
({max_penjualan['jumlah_terjual']} unit)")
print(f'Barang Penjualan Terendah: {min_penjualan['nama_barang']}
({min_penjualan['jumlah_terjual']} unit)")

```



```

[ ] try:
    # Memastikan kolom numerik
    penjualan['jumlah_terjual'] = pd.to_numeric(penjualan['jumlah_terjual'], errors='coerce')
    penjualan['harga_jual'] = pd.to_numeric(penjualan['harga_jual'], errors='coerce')

    # Menghitung pendapatan
    penjualan['pendapatan'] = penjualan['jumlah_terjual'] * penjualan['harga_jual']
    total_terjual = penjualan['jumlah_terjual'].sum()
    total_pendapatan = penjualan['pendapatan'].sum()

    # Menemukan barang dengan penjualan tertinggi dan terendah
    max_penjualan = penjualan.loc[penjualan['jumlah_terjual'].idxmax()]
    min_penjualan = penjualan.loc[penjualan['jumlah_terjual'].idxmin()]

    print(f'Total Unit Terjual: {total_terjual:,} unit")
    print(f'Total Pendapatan: Rp {total_pendapatan:,.2f}")
    print(f'Barang Penjualan Tertinggi: {max_penjualan['nama_barang']} ({max_penjualan['jumlah_terjual']:.0f} unit)")
    print(f'Barang Penjualan Terendah: {min_penjualan['nama_barang']} ({min_penjualan['jumlah_terjual']:.0f} unit)")
except KeyError as e:
    print(f"Error: Kolom {e} tidak ditemukan di tabel Penjualan.")
except Exception as e:
    print(f"Error: {e}")

```

Output

```

Total Unit Terjual: 450 unit
Total Pendapatan: Rp 6,850,000.00
Barang Penjualan Tertinggi: Gula Pasir 1kg (100 unit)
Barang Penjualan Terendah: Sabun Cair 250ml (10 unit)

```

4. Margin Keuntungan Sederhana:

- Hitung margin keuntungan dengan menggabungkan data Produksi dan Penjualan berdasarkan kode_barang.

Kode Python

Menghitung biaya produksi per unit

```
produksi['biaya_per_unit'] = produksi['biaya_produksi'] / produksi['jumlah_produksi']
```

Menggabungkan tabel penjualan dan produksi

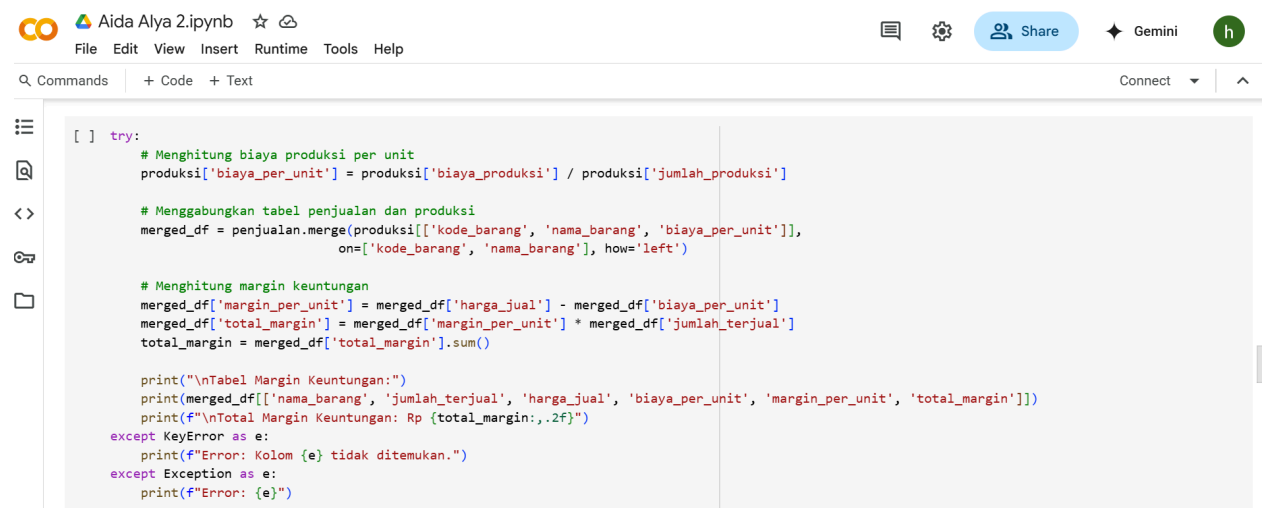
```
merged_df = penjualan.merge(produksi[['kode_barang', 'biaya_per_unit']],
on='kode_barang')
```

Menghitung margin keuntungan per unit dan total margin

```
merged_df['margin_per_unit'] = merged_df['harga_jual'] - merged_df['biaya_per_unit']
merged_df['total_margin'] = merged_df['margin_per_unit'] * merged_df['jumlah_terjual']
total_margin = merged_df['total_margin'].sum()
```

Menampilkan tabel margin

```
print("\nTabel Margin Keuntungan:")
print(merged_df[['nama_barang', 'jumlah_terjual', 'harga_jual', 'biaya_per_unit',
'margin_per_unit', 'total_margin']])
print(f"\nTotal Margin Keuntungan: Rp {total_margin:,.2f}")
```



```
[ ] try:
# Menghitung biaya produksi per unit
produksi['biaya_per_unit'] = produksi['biaya_produksi'] / produksi['jumlah_produksi']

# Menggabungkan tabel penjualan dan produksi
merged_df = penjualan.merge(produksi[['kode_barang', 'nama_barang', 'biaya_per_unit']],
on=['kode_barang', 'nama_barang'], how='left')

# Menghitung margin keuntungan
merged_df['margin_per_unit'] = merged_df['harga_jual'] - merged_df['biaya_per_unit']
merged_df['total_margin'] = merged_df['margin_per_unit'] * merged_df['jumlah_terjual']
total_margin = merged_df['total_margin'].sum()

print("\nTabel Margin Keuntungan:")
print(merged_df[['nama_barang', 'jumlah_terjual', 'harga_jual', 'biaya_per_unit', 'margin_per_unit', 'total_margin']])
print(f"\nTotal Margin Keuntungan: Rp {total_margin:,.2f}")
except KeyError as e:
print(f"Error: Kolom {e} tidak ditemukan.")
except Exception as e:
print(f"Error: {e}")
```

Output

Tabel Margin Keuntungan:				
	nama_barang	jumlah_terjual	harga_jual	biaya_per_unit \
0	Kopi Bubuk 500g	50	15000.0	5000.0
1	Teh Celup 50s	20	25000.0	4000.0
2	Gula Pasir 1kg	100	12000.0	5000.0
3	Minyak Goreng 1L	50	20000.0	5000.0
4	Beras Premium 5kg	20	60000.0	5000.0
5	Susu Bubuk 400g	30	30000.0	5000.0
6	Tepung Terigu 1kg	50	10000.0	5000.0
7	Mie Instan 70g	100	3000.0	2000.0
8	Sarden Kaleng 155g	20	15000.0	4000.0
9	Sabun Cair 250ml	10	20000.0	5000.0
	margin_per_unit	total_margin		
0	10000.0	500000.0		
1	21000.0	420000.0		
2	7000.0	700000.0		
3	15000.0	750000.0		
4	55000.0	1100000.0		
5	25000.0	750000.0		
6	5000.0	250000.0		
7	1000.0	100000.0		
8	11000.0	220000.0		
9	15000.0	150000.0		
Total Margin Keuntungan: Rp 4,940,000.00				

Langkah 4: Visualisasi Data

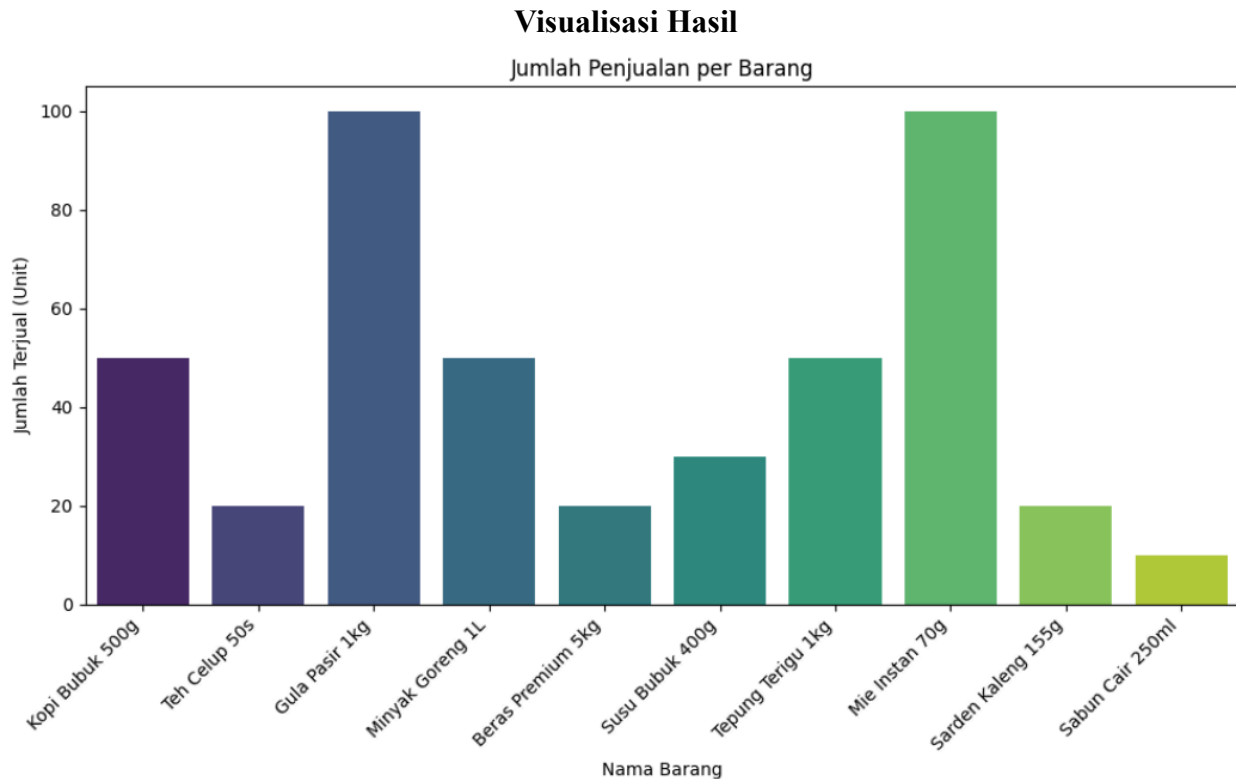
Kita akan membuat grafik batang untuk menampilkan jumlah terjual per barang, mirip dengan grafik sebelumnya.

Kode Python

Grafik batang jumlah terjual per barang

```
plt.figure(figsize=(10, 6))
sns.barplot(x='nama_barang', y='jumlah_terjual', data=penjualan, palette='viridis')
plt.title('Jumlah Penjualan per Barang')
plt.xlabel('Nama Barang')
plt.ylabel('Jumlah Terjual (Unit)')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
[ ] try:
    plt.figure(figsize=(10, 6))
    sns.barplot(x='nama_barang', y='jumlah_terjual', data=penjualan, palette='viridis')
    plt.title('Jumlah Penjualan per Barang')
    plt.xlabel('Nama Barang')
    plt.ylabel('Jumlah Terjual (Unit)')
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()
except KeyError as e:
    print(f"Error: Kolom {e} tidak ditemukan.")
except Exception as e:
    print(f"Error: {e}")
```



Langkah 5: Menyimpan Hasil Analisis

Jika Anda ingin menyimpan hasil analisis (misalnya tabel margin keuntungan) ke file CSV:

Kode Python

```
# Menyimpan tabel margin ke CSV
merged_df.to_csv('analisis_margin.csv', index=False)
files.download('analisis_margin.csv')
```

Langkah 6: Menjalankan Kode

1. Jalankan setiap sel kode secara berurutan dengan menekan **Shift + Enter** di Google Colab.
2. Pastikan file CSV sudah diunggah sebelum menjalankan sel yang membaca data.
3. Periksa output di bawah setiap sel untuk memastikan hasilnya sesuai (misal: total produksi, rata-rata stok, grafik).

Kode Lengkap

Berikut adalah kode lengkap yang bisa Anda salin ke Google Colab:

python

Copy

```
# Langkah 1: Impor library
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# Langkah 2: Unggah dan baca file CSV
```

```
from google.colab import files
```

```
uploaded = files.upload()
```

```
# Membaca file CSV
```

```
produksi = pd.read_csv('produksi.csv')
```

```
persediaan = pd.read_csv('persediaan.csv')
```

```
penjualan = pd.read_csv('penjualan.csv')
```

```
# Menampilkan beberapa baris untuk memastikan data terbaca
```

```
print("Tabel Produksi:")
```

```
print(produksi.head())
```

```
print("\nTabel Persediaan:")
```

```
print(persediaan.head())
```

```
print("\nTabel Penjualan:")
```

```
print(penjualan.head())
```

```
# Langkah 3: Analisis Data
```

```
# 3.1 Total Produksi dan Biaya Produksi
```

```
total_produksi = produksi['jumlah_produksi'].sum()
```

```
total_biaya_produksi = produksi['biaya_produksi'].sum()
```

```
print(f"\nTotal Jumlah Produksi: {total_produksi} unit")
```

```
print(f"Total Biaya Produksi: Rp {total_biaya_produksi:,.2f}")
```

3.2 Rata-rata Stok

```
total_stok = persediaan['jumlah_stok'].sum()
```

```
rata_rata_stok = total_stok / len(persediaan)
```

```
print(f"\nTotal Stok: {total_stok} unit")
```

```
print(f"Rata-rata Stok per Barang: {rata_rata_stok:.2f} unit")
```

3.3 Tingkat Penjualan

```
penjualan['pendapatan'] = penjualan['jumlah_terjual'] * penjualan['harga_jual']
```

```
total_terjual = penjualan['jumlah_terjual'].sum()
```

```
total_pendapatan = penjualan['pendapatan'].sum()
```

```
max_penjualan = penjualan.loc[penjualan['jumlah_terjual'].idxmax()]
```

```
min_penjualan = penjualan.loc[penjualan['jumlah_terjual'].idxmin()]
```

```
print(f"\nTotal Unit Terjual: {total_terjual} unit")
```

```
print(f"Total Pendapatan: Rp {total_pendapatan:,.2f}")
```

```
print(f"Barang Penjualan Tertinggi: {max_penjualan['nama_barang']}  
( {max_penjualan['jumlah_terjual']} unit)")
```

```
print(f"Barang Penjualan Terendah: {min_penjualan['nama_barang']}  
( {min_penjualan['jumlah_terjual']} unit)")
```

3.4 Margin Keuntungan

```
produksi['biaya_per_unit'] = produksi['biaya_produksi'] / produksi['jumlah_produksi']
```

```
merged_df = penjualan.merge(produksi[['kode_barang', 'biaya_per_unit']], on='kode_barang')
```

```
merged_df['margin_per_unit'] = merged_df['harga_jual'] - merged_df['biaya_per_unit']
```

```
merged_df['total_margin'] = merged_df['margin_per_unit'] * merged_df['jumlah_terjual']
```

```
total_margin = merged_df['total_margin'].sum()
```

```
print("\nTabel Margin Keuntungan:")
```

```
print(merged_df[['nama_barang', 'jumlah_terjual', 'harga_jual', 'biaya_per_unit',  
'margin_per_unit', 'total_margin']])
```

```
print(f"\nTotal Margin Keuntungan: Rp {total_margin:,.2f}")
```

Langkah 4: Visualisasi

```
plt.figure(figsize=(10, 6))
sns.barplot(x='nama_barang', y='jumlah_terjual', data=penjualan, palette='viridis')
plt.title('Jumlah Penjualan per Barang')
plt.xlabel('Nama Barang')
plt.ylabel('Jumlah Terjual (Unit)')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

Langkah 5: Menyimpan hasil

```
merged_df.to_csv('analisis_margin.csv', index=False)
files.download('analisis_margin.csv')
```