

ANALISI DAN VISUALISASI DATA LAPORAN PENJUALAN TOKO ONLINE MELALUI PENDEKATAN DATA SCIENCE MENGGUNAKAN PYTHON

Ray Valentino Pradana

Universitas Duta Bangsa Surakarta

Email:210103033@mhs.udb.ac.id

Abstrak : Dalam konteks analisis data laporan penjualan toko online, pendekatan Data Science menggunakan bahasa pemrograman Python dan pustaka seperti Pandas, Matplotlib, dan NumPy menjadi kunci penting. Dalam laporan ini, kami menjelajahi bagaimana Python dan alat-alat ini digunakan untuk mengolah, menganalisis, dan memvisualisasikan data penjualan. Pendekatan ini memungkinkan kita untuk menggali wawasan berharga, termasuk pemahaman tentang tren produk, profil pelanggan, dan kinerja penjualan. Melalui visualisasi data yang kuat, kami mempresentasikan temuan kami secara informatif. Dengan kata kunci Python, Pandas, Matplotlib, NumPy, laporan penjualan, dan visualisasi data, laporan ini menggambarkan pentingnya penggunaan alat-alat ini dalam menganalisis dan memahami data penjualan dalam lingkungan perdagangan daring yang kompetitif.

Kata kunci: Python,Pandas,Matplotlib,Numpy,Sales Report,Data Visualization

Abstract: In the context of analyzing online store sales reports, a Data Science approach using the Python programming language and libraries such as Pandas, Matplotlib, and NumPy is of vital importance. In this report, we explore how Python and these tools are used to process, analyze, and visualize sales data. This approach enables us to unearth valuable insights, including an understanding of product trends, customer profiles, and sales performance. Through powerful data visualization, we present our findings informatively. With keywords such as Python, Pandas, Matplotlib, NumPy, sales reports, and data visualization, this report highlights the significance of using these tools for analyzing and comprehending sales data in a competitive online trade environment.

Keywords: At least 3 words and a maximum of 6 words, (first word; second word; third word)

Pendahuluan

Ilmu data alias data science merupakan fokus studi yang berhubungan dengan besarnya volume data menggunakan teknik modern demi menemukan pola-pola tak terlihat, mendapat informasi bermakna, serta membuat keputusan bisnis dengan informasi tersebut.[1]

Di era digital saat ini, toko online telah menjadi salah satu aspek penting dalam dunia perdagangan. Semakin banyak perusahaan dan individu yang beralih ke platform online untuk menjual produk dan layanan mereka. Dalam konteks ini, data menjadi harta karun yang tak ternilai harganya. Data penjualan toko online dapat memberikan wawasan

berharga tentang perilaku pelanggan, tren produk, kinerja penjualan, dan banyak informasi penting lainnya.

Dalam upaya untuk memahami dan memanfaatkan data ini, Data Science menjadi pendekatan yang sangat penting. Data Science menggabungkan teknik analisis data, pemodelan statistik, dan pemrograman komputer untuk menggali wawasan dari data yang ada. Python, sebagai salah satu bahasa pemrograman yang paling populer dalam komunitas Data Science, menawarkan banyak pustaka dan alat yang kuat untuk menganalisis dan memvisualisasikan data.

Laporan ini akan menjadi panduan praktis yang membantu Anda memahami bagaimana Data Science dan Python dapat digunakan untuk mendekati dan memahami data penjualan toko online. Dengan demikian, Anda dapat memanfaatkan informasi yang ditemukan dalam data ini untuk mengambil keputusan yang lebih cerdas, meningkatkan efisiensi operasional, dan meraih kesuksesan lebih besar dalam lingkungan perdagangan daring yang kompetitif.

Landasan Teori

Penelitian ini didasarkan pada konsep-konsep dasar dalam visualisasi data dan penggunaan Python sebagai alat untuk melakukan visualisasi data. Berikut ini adalah landasan teori yang digunakan dalam artikel ini:

1. Visualisasi Data

Visualisasi data adalah proses menggunakan elemen visual seperti diagram, grafik, atau peta untuk merepresentasikan data. Visualisasi data menerjemahkan yang kompleks, bervolume tinggi, atau numerik menjadi representasi visual yang lebih mudah diproses. Alat visualisasi data meningkatkan dan mengotomatiskan proses komunikasi visual untuk mendapatkan akurasi dan detail. Anda dapat menggunakan representasi visual untuk mengekstraksi wawasan yang dapat ditindaklanjuti dari data mentah.[2]

2. Pendekatan Data Science dalam Visualisasi Data

Pendekatan Data Science dalam Visualisasi Data merupakan salah satu aspek penting dalam proses analisis data. Visualisasi data adalah teknik untuk mewakili informasi dan wawasan dari data dalam bentuk grafis, seperti grafik, grafik batang, grafik garis, peta panas, dan banyak lagi. Pendekatan Data Science dalam visualisasi data memainkan peran kunci dalam memahami, menggali, dan berkomunikasi wawasan yang ditemukan dalam data. Berikut adalah beberapa poin penting dalam pendekatan Data Science terhadap visualisasi data

3. Python

Python untuk Data Science mencakup sejumlah konsep esensial. Python, sebagai bahasa pemrograman utama dalam Data Science, dikenal dengan sintaksis

yang bersih dan mudah dimengerti. Python juga didukung oleh beragam pustaka yang penting, seperti NumPy dan pandas, yang memfasilitasi analisis dan pengolahan data. Pemahaman tentang struktur data seperti Data Frame, Series, dan NumPy Array menjadi landasan dalam pengelolaan data tabular. Kemampuan pengindeksan dan pemotongan data memungkinkan akses data dengan efisien. Selain itu, pemrosesan data dengan bantuan pandas melibatkan berbagai operasi seperti pemfilteran dan penggabungan data. Visualisasi data dengan pustaka seperti matplotlib dan seaborn penting dalam komunikasi wawasan dari data. Pemahaman statistik, analisis data, dan pembelajaran mesin memberikan alat untuk menggali wawasan lebih dalam.

4. Pandas

Pandas adalah pustaka Python yang penting dalam analisis dan manipulasi data tabular. Dengan fokus pada struktur data seperti DataFrame dan Series, Pandas memfasilitasi pengolahan data yang mudah dan efisien. Nama "Pandas" berasal dari "Panel Data," dan pustaka ini digunakan secara luas dalam berbagai bidang yang melibatkan data.

Fitur-Fitur Pandas:

- 1) DataFrame dan Series: DataFrame adalah struktur data utama yang mirip dengan tabel, sedangkan Series adalah struktur data satu dimensi yang komponen dari DataFrame.
- 2) Pengindeksan dan Pemotongan: Memungkinkan akses data dengan mengindeks dan memotong data dengan mudah.
- 3) Pemfilteran dan Pemilihan Data: Menyederhanakan pemfilteran dan pemilihan data berdasarkan kriteria tertentu.
- 4) Pemrosesan Data: Menyediakan fungsi dan metode untuk melakukan pemrosesan data, seperti penggabungan data, pengelompokan, pengurutan, dan transformasi.
- 5) Manipulasi Tanggal dan Waktu: Mendukung data tanggal dan waktu dengan baik, memudahkan analisis deret waktu.

5. Numpy

NumPy adalah pustaka Python yang fundamental dalam pemrosesan data numerik. Singkatan dari "Numerical Python," NumPy memberikan landasan yang kuat untuk pemrosesan dan analisis data berbasis matriks dan array. Ini digunakan secara luas dalam komputasi ilmiah, statistik, dan Data Science.

Fitur-Fitur NumPy:

- 1) Array Multidimensi: NumPy memungkinkan pembuatan dan manipulasi array multidimensi yang efisien, yang merupakan struktur data utama dalam pustaka ini.

- 2) Operasi Matematika: Menyediakan beragam fungsi dan operasi matematika yang kuat untuk mengolah data numerik, termasuk operasi elemen demi elemen pada array.
- 3) Indeks dan Slicing: Memungkinkan indeks dan pemotongan data yang efisien dalam array, serupa dengan list Python.
- 4) Broadcasting: NumPy memiliki fitur "broadcasting" yang memungkinkan operasi antara array dengan ukuran yang berbeda tanpa perlu mengekspansinya.
- 5) Fungsi Statistik: Menyediakan berbagai fungsi statistik, seperti mean, median, standar deviasi, dan lainnya.

6. Matplotlib

Matplotlib adalah pustaka Python yang penting dalam visualisasi data. Matplotlib memungkinkan pengguna untuk membuat berbagai jenis grafik dan visualisasi data yang informatif. Ini adalah salah satu alat utama dalam menggambarkan hasil analisis data dan berkomunikasi wawasan dari data.

Fitur-Fitur Matplotlib:

- 1) Beragam Jenis Grafik: Matplotlib mendukung berbagai jenis grafik, termasuk grafik garis, grafik batang, diagram lingkaran, peta panas, scatter plot, dan banyak lagi.
- 2) Kustomisasi Visual: Memungkinkan kustomisasi grafik seperti warna, legenda, label, skala, dan sebagainya untuk menciptakan visualisasi yang menarik dan informatif.
- 3) Subplot dan Multiplot: Anda dapat membuat subplot atau multiplot dalam satu gambar untuk membandingkan atau menampilkan lebih dari satu grafik dalam satu tampilan.
- 4) Interaktivitas: Matplotlib mendukung interaktivitas dengan memungkinkan pengguna untuk menambahkan elemen interaktif ke grafik, seperti zoom, panning, dan penjelasan tambahan.
- 5) Ekspor Grafik: Anda dapat menyimpan grafik dalam berbagai format file seperti PNG, JPEG, SVG, dan PDF.

Metode Penelitian

Langkah pertama dalam melakukan visualisasi data adalah mengumpulkan data yang akan digunakan. Dalam kasus ini, akan menggunakan data penjualan toko online dalam bentuk CSV. Data ini berisi informasi tentang penjualan produk dalam periode waktu tertentu, termasuk jenis produk, harga, jumlah terjual, dan tanggal penjualan, nama customer, jenis kelamin customer, kota customer, cabang toko dan jenis pembayaran.

Metode Penelitian merupakan proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk menyelesaikan masalah dengan dengan mengadakan studi langsung kelapangan untuk mengumpulkan data. Adapun metode dalam penelitian ini mencakup :

Teknik Pengumpulan Data Teknik pengumpulan data yang digunakan dalam analisis data Daftar Pemilih Tetap (DPT) di Karesidenan Surakarta Pemilu 2024 menggunakan pengumpulan data studi dokumen. Berikut uraian yang digunakan :

- 1) Primer Upaya dalam menganalisis data ini menggunakan pemograman python yang di dalamnya menggunakan library pandas, numpy, dan matplotlib.
- 2) Sekunder Metode Pengumpulan data ini digunakan untuk mengumpulkan data yang ada dalam situs Komisi Pemilihan Umum (KPU).

Hasil dan Pembahasan

Pada penelitian ini, dataset yang digunakan adalah data penjualan toko online pada sebuah marketplace yang disimpan dalam format CSV (Comma-Separated Values). Dataset ini berisi informasi tentang transaksi penjualan seperti tanggal transaksi, produk yang dibeli, harga produk, jumlah produk yang dibeli, dan kategori produk. Dataset ini terdiri dari 1000 baris data dan 8 kolom data. Kolom data yang terdapat dalam dataset adalah sebagai berikut:

Tabel 1. Data set yang digunakan

No	Kolom	Keterangan
1	Invoice ID	Kode Order Pelangan
2	Branch	Cabang Toko
3	City	Kota Pembeli
4	Customer type	Jenis Customer pelanggan atau member
5	Gender	Jenis Kelamin Pelangan
6	Product line	Jenis Produk
7	Unit price	Harga Perbarang
8	Quantity	Jumlah Barang
9	Tax 5%	Pajak 5%
10	Total	Total Transaksi
11	Date	Tanggal
12	Time	Waktu
13	Payment	Jenis Pembayaran
14	cogs	Harga Pokok Barang
15	gross margin percentage	selisih antara pendapatan dan harga pokok penjualan
16	gross income	Total pendapatan
17	Rating	Penilaian produk oleh pelanggan

IMPLEMENTASI DAN PENGUJIAN

1. Pandas

Berikut ini adalah implementasi dan pengujian menggunakan library pandas.

- 1) Menampilkan semua dataset file csv tetapi disini dibatasi menggunakan paramater.

Source code:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('supermarket_sales.csv')
print('menampilkan semua data')
df = pd.read_csv('supermarket_sales.csv')
print(df.head(10))
```

Gambar 1. Menampilkan semua data laporan penjualan

Output:

```
... menampilkan semua data
```

	Invoice ID	Branch	City	Customer type	Gender	\
0	750-67-8428	A	Yangon	Member	Female	
1	226-31-3081	C	Naypyitaw	Normal	Female	
2	631-41-3108	A	Yangon	Normal	Male	
3	123-19-1176	A	Yangon	Member	Male	
4	373-73-7910	A	Yangon	Normal	Male	
5	699-14-3026	C	Naypyitaw	Normal	Male	
6	355-53-5943	A	Yangon	Member	Female	
7	315-22-5665	C	Naypyitaw	Normal	Female	
8	665-32-9167	A	Yangon	Member	Female	
9	692-92-5582	B	Mandalay	Member	Female	

	Product line	Unit price	Quantity	Tax 5%	Total	Date	\
0	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	
1	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	
2	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	
3	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	
4	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	
5	Electronic accessories	85.39	7	29.8865	627.6165	3/25/2019	
6	Electronic accessories	68.84	6	20.6520	433.6920	2/25/2019	
7	Home and lifestyle	73.56	10	36.7800	772.3800	2/24/2019	
8	Health and beauty	36.26	2	3.6260	76.1460	1/10/2019	
9	Food and beverages	54.84	3	8.2260	172.7460	2/20/2019	


```
...
6 14:36 Ewallet 413.04 4.761905 20.6520 5.8
7 11:30 Ewallet 735.60 4.761905 36.7800 8.0
8 17:15 Credit card 72.52 4.761905 3.6260 7.2
9 13:27 Credit card 164.52 4.761905 8.2260 5.9
```

Gambar 2. Output Menampilkan semua data laporan penjualan

- 2) Menampilkan data laporan penjualan dengan memanggil kolom City dan juga menggunakan metode sorting ascending.

Source code:

```

# Ascending
df["City"] = df["City"].astype("string")
daerah = df[df['City'] == 'Yangon']

daerah = daerah.sort_values(by=['City'], ascending=True)
print('\nMenampilkan Daftar penjualan di kota Yangon\n')
print(daerah.head(10))

```

Gambar 3. Menampilkan semua laporan penjualan di kota Yangon

Output:

Menampilkan Daftar penjualan di kota Yangon							
	Invoice ID	Branch	City	Customer type	Gender	Product line	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	
689	305-03-2383	A	Yangon	Normal	Female	Food and beverages	
687	648-83-1321	A	Yangon	Member	Male	Home and lifestyle	
683	105-10-6182	A	Yangon	Member	Male	Fashion accessories	
679	130-67-4723	A	Yangon	Member	Male	Food and beverages	
677	274-05-5470	A	Yangon	Member	Female	Food and beverages	
674	811-03-8790	A	Yangon	Normal	Female	Electronic accessories	
670	588-47-8641	A	Yangon	Member	Male	Fashion accessories	
665	518-17-2983	A	Yangon	Normal	Female	Fashion accessories	
659	491-38-3499	A	Yangon	Member	Male	Fashion accessories	
	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment
0	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet
689	67.10	3	10.0650	211.3650	2/15/2019	10:36	Cash
687	63.56	10	31.7800	667.3800	1/16/2019	17:59	Cash
683	21.48	2	2.1480	45.1080	2/27/2019	12:22	Ewallet
679	48.50	6	14.5500	305.5500	1/11/2019	13:57	Ewallet
677	73.47	4	14.6940	308.5740	2/23/2019	18:30	Cash
674	45.48	10	22.7400	477.5400	3/1/2019	10:22	Credit card
670	56.04	10	28.0200	588.4200	1/14/2019	19:30	Ewallet
665	48.63	4	9.7260	204.2460	2/4/2019	15:44	Ewallet
...							
674	454.80		4.761905	22.7400		4.8	
670	560.40		4.761905	28.0200		4.4	
665	194.52		4.761905	9.7260		7.6	
659	55.45		4.761905	2.7725		4.9	

Gambar 4. Output menampilkan semua laporan penjualan di kota Yangon

- Menampilkan laporan penjualan jumlah laki-laki terbanyak dan terkecil dengan memanggil index City, Gender dan memanggil kolom Male .

Source code:

```

hitung_City_jenis_kelamin = df[df['Gender'] == 'Male']
hitung_City_jenis_kelamin = hitung_City_jenis_kelamin.groupby(['City',
'Gender']).size().reset_index(name='Jumlah')
kota_terbanyak = hitung_City_jenis_kelamin.nlargest(1, 'Jumlah')
kota_terkecil = hitung_City_jenis_kelamin.nsmallest(1, 'Jumlah')

print('\nKota dengan Gender "Pria" Terbanyak:')
print(kota_terbanyak)

print('\nKota dengan Gender "Pria" Terkecil:')
print(kota_terkecil)

```

Gambar 5. Menampilkan laporan penjualan Laki-Laki
Terbanyak dan Terkecil di Kota Yangon

Output :

```
Kota dengan Gender "Pria" Terbanyak:
  City Gender  Jumlah
2  Yangon   Male    179

Kota dengan Gender "Pria" Terkecil:
  City Gender  Jumlah
1  Naypyitaw Male    150
```

Gambar 6. Hasil Output Menampilkan laporan penjualan Laki-Laki Terbanyak dan Terkecil di kota Yangon

- 4) Menampilkan laporan penjualan jumlah laki-laki terbanyak dan terkecil dengan memanggil index City, Gender dan memanggil kolom Female .

Source code:

```
hitung_City_jenis_kelamin = df[df['Gender'] == 'Female']
hitung_City_jenis_kelamin = hitung_City_jenis_kelamin.groupby(['City',
'Gender']).size().reset_index(name='Jumlah')
kota_terbanyak = hitung_City_jenis_kelamin.nlargest(1, 'Jumlah')
kota_terkecil = hitung_City_jenis_kelamin.nsmallest(1, 'Jumlah')
print('\nKota dengan Gender "Wanita" Terbanyak:')
print(kota_terbanyak)

print('\nKota dengan Gender "Wanita" Terkecil:')
print(kota_terkecil)
```

Gambar 7. Menampilkan laporan penjualan perempuan Terbanyak dan Terkecil di Kota Yangon

Output:

```
Kota dengan Gender "Wanita" Terbanyak:
  City Gender  Jumlah
1  Naypyitaw Female    178

Kota dengan Gender "Wanita" Terkecil:
  City Gender  Jumlah
2  Yangon   Female    161
```

Gambar 8. Hasil Output Menampilkan laporan penjualan perempuan Terbanyak dan Terkecil di kota Yangon

- 5) Mengurutkan jenis produk yang terjual paling banyak dan sedikit berdasarkan kota
- Source code:


```
# Kelompokkan data berdasarkan 'City' dan 'Product line', lalu hitung jumlahnya
hitung_kota_produk = df.groupby(['City', 'Product line']).size().reset_index(name='Jumlah')

# Temukan jumlah maksimum dan minimum untuk setiap kota
jumlah_maksimum = hitung_kota_produk.groupby('City')['Jumlah'].max()
jumlah_minimum = hitung_kota_produk.groupby('City')['Jumlah'].min()

# Filter baris dengan jumlah maksimum dan minimum untuk setiap kota
produk_terbanyak = hitung_kota_produk[hitung_kota_produk.apply(lambda baris: baris['Jumlah'] ==
jumlah_maksimum[baris['City']], axis=1)]
produk_terkecil = hitung_kota_produk[hitung_kota_produk.apply(lambda baris: baris['Jumlah'] ==
jumlah_minimum[baris['City']], axis=1)]

print('\nJenis Produk Terkecil Berdasarkan Kota:')
print(produk_terbanyak)

print('\nJenis Produk Terbanyak Berdasarkan Kota:')
print(produk_terkecil)
```

Gambar 9. Menampilkan jenis produk yang terjual paling banyak dan sedikit berdasarkan kota

Output:

Jenis Produk Terkecil Berdasarkan Kota:			
	City	Product line	Jumlah
1	Mandalay	Fashion accessories	62
5	Mandalay	Sports and travel	62
8	Naypyitaw	Food and beverages	66
16	Yangon	Home and lifestyle	65

Jenis Produk Terbanyak Berdasarkan Kota:			
	City	Product line	Jumlah
2	Mandalay	Food and beverages	50
4	Mandalay	Home and lifestyle	50
10	Naypyitaw	Home and lifestyle	45
11	Naypyitaw	Sports and travel	45
15	Yangon	Health and beauty	47

Gambar 10. Hasil menampilkan jenis produk yang terjual paling banyak dan sedikit berdasarkan kota

2. Numpy

Berikut ini adalah implementasi dan pengujian menggunakan library numpy di batasi hanya menghitung 10 data.

- 1) Menampilkan Data Operasi Aritmatika yang meliputi penjumlahan, pengurangan, perkalian, dan pembagian. Tetapi tipe data yang ada dalam csv tidak ada/object maka harus dikonversi terlebih dahulu ke tipe data numerik.

Source code :

```

import pandas as pd
import numpy as np

df = pd.read_csv('supermarket_sales.csv')
n = 10

# Mengasumsikan 'City' adalah kolom yang benar
df["City"] = df["City"].astype("string")

# Mengambil data Unit Price dan Quantity dari DataFrame ke dalam np.array
unit_price = df['Unit price'].to_numpy()
quantity = df['Quantity'].to_numpy()

# Melakukan operasi penjumlahan, pengurangan, perkalian, dan pembagian dengan np.array
penjumlahan = unit_price + quantity
pengurangan = unit_price - quantity
perkalian = unit_price * quantity

# Pastikan tidak ada pembagian oleh nol
pembagian = np.divide(unit_price, quantity, out=np.zeros_like(unit_price), where=quantity!=0)

print("\nOperasi Aritmatika menggunakan Data Unit Price dan Quantity di Seluruh DataFrame (10 Data Pertama)")
print("\nPenjumlahan")
print(penjumlahan[:n])
print("\nPengurangan")
print(pengurangan[:n])
print("\nPerkalian")
print(perkalian[:n])
print("\nPembagian")
print(pembagian[:n])

```

Gambar 11. Menampilkan Data Operasi Aritmatika yang meliputi penjumlahan, pengurangan, perkalian, dan pembagian.

Output:

```

Operasi Aritmatika menggunakan Data Unit Price dan Quantity di Seluruh DataFrame (10 Data Pertama)

Penjumlahan
[81.69 20.28 53.33 66.22 93.31 92.39 74.84 83.56 38.26 57.84]

Pengurangan
[67.69 10.28 39.33 50.22 79.31 78.39 62.84 63.56 34.26 51.84]

Perkalian
[522.83 76.4 324.31 465.76 604.17 597.73 413.04 735.6 72.52 164.52]

Pembagian
[10.67      3.056      6.61857143  7.2775      12.33      12.19857143
 11.47333333  7.356      18.13      18.28      ]

```

Gambar 12. Output Operasi Aritmatika yang meliputi penjumlahan, pengurangan, perkalian, dan pembagian.

- 2) Menampilkan Data Transpose, Reshape, dan Flatten. yang data array yang dihasilkan yaitu matrik 1 Dimensi.

Source code:

```

# Hasil data matrix array hanya 1 dimensi

tup = unit_price.reshape(-1, 1).astype(int)
tq = quantity.reshape(-1, 1).astype(int)
print("\nTranspose dari Data Unit Price ")
print(tup[:n])
print("\nTranspose dari Data Quantity ")
print(tq[:n])

rsup = unit_price.reshape(-1, 1).astype(int)
rsq = quantity.reshape(-1, 1).astype(int)
print("\nReshape dari Data Unit Price ")
print(rsup[:n])
print("Reshape dari Data Quantity ")
print(rsq[:n])

fup = unit_price.flatten().astype(int)
fq = quantity.flatten().astype(int)
print("\nFlatten dari Data Unit Price ")
print(fup[:n])
print("\nFlatten dari Data Quantity ")
print(fq[:n])

```

Gambar 13. Menampilkan Data Transpose, Reshape, dan Flatten.

Output :

```

...
Transpose dari Data Unit Price
[[74]
 [15]
 [46]
 [58]
 [86]
 [85]
 [68]
 [73]
 [36]
 [54]]

Transpose dari Data Quantity
[[ 7]
 [ 5]
 [ 7]
 [ 8]
 [ 7]
 [ 7]
 [ 6]
 [10]
 [ 2]
 [ 3]]
...
[74 15 46 58 86 85 68 73 36 54]

Flatten dari Data Quantity
[ 7 5 7 8 7 7 6 10 2 3]

```

Gambar 14. Output menampilkan Data Transpose, Reshape, dan Flatten.

Menampilkan Data Min, Max, Rata-rata, Sum, dan Standar Deviasi. Memanggil data unite price dan quantity.

Source code:



Gambar 15. Menampilkan Data Min, Max, Rata-rata, Sum, dan Standar Deviasi.

Output:

```

Jumlah data unit price (10 data pertama)
Nilai Minimal: 15.28
Nilai Maksimal: 86.31
Nilai Rata-rata: 59.972
Total Nilai: 599.72
Standar Deviasi: 21.435495235706593

Jumlah data quantity (10 data pertama)
Nilai Minimal: 2
Nilai Maksimal: 10
Nilai Rata-rata: 6.2
Total Nilai: 62
Standar Deviasi: 2.227105745132009

```

Gambar 16. Output menampilkan Data Min, Max, Rata-rata, Sum, dan Standar Deviasi.

- 3) Menampilkan Data Min, Max, Rata-rata, Sum, dan Standar Deviasi. Memanggil data unite price dan quantity.

Source code:



Gambar 15. Menampilkan Data Min, Max, Rata-rata, Sum, dan Standar Deviasi.

Output:

```
Jumlah data unit price (10 data pertama)
Nilai Minimal: 15.28
Nilai Maksimal: 86.31
Nilai Rata-rata: 59.972
Total Nilai: 599.72
Standar Deviasi: 21.435495235706593

Jumlah data quantity (10 data pertama)
Nilai Minimal: 2
Nilai Maksimal: 10
Nilai Rata-rata: 6.2
Total Nilai: 62
Standar Deviasi: 2.227105745132009
```

Gambar 16. Output menampilkan Data Min, Max, Rata-rata, Sum, dan Standar Deviasi.

3. Matplotlib

Berikut ini adalah implementasi dan pengujian menggunakan library matplotlib yang di batasi hanya menghitung dan menampilkan 50 data.

- 1) Count plot yang menunjukan jenis pembayaran pada setiap produk yang akan memanggil index Product line dan Payment

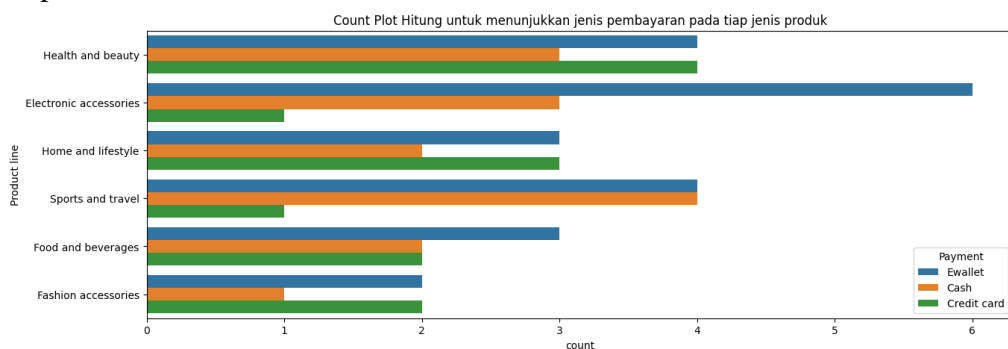
Source code:

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

df = pd.read_csv('supermarket_sales.csv')
n = 50
## countplot ##
plt.figure(figsize=(15, 5))
sns.countplot(y='Product line', data=df.iloc[:n], hue='Payment').set(
    title='Count Plot Hitung untuk menunjukkan jenis pembayaran pada tiap jenis produk')
```

Gambar 17. Menampilkan count plot jenis pembayaran pada setiap produk

Output:



Gambar 18. Output count plot jenis pembayaran pada setiap produk

- 2) Menampilkan pie chart yang menunjukan banyaknya produk yang di beli berdasarkan jenis produk yang akan memanggil index Product line.

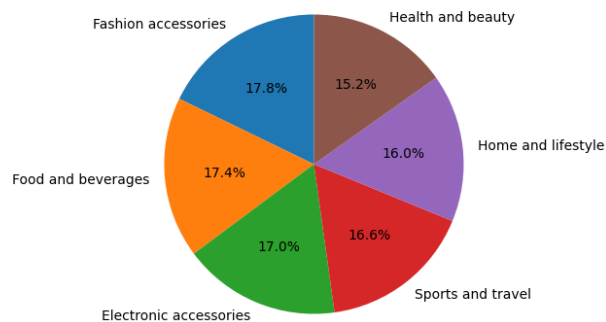
Source code:

```
## pie ##
category_counts = df['Product line'].value_counts()[:n]
plt.figure(figsize=(10, 5))
plt.pie(category_counts, labels=category_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('Pie Chart menunjukkan banyaknya produk yang di beli berdasarkan jenis produk')
plt.show()
```

Gambar 19. Menampilkan pie chart meunjukan banyaknya produk yang di beli

Output:

Pie Chart menunjukkan banyaknya produk yang di beli berdasarkan jenis produk



Gambar 20. Output pie chart meunjukan banyaknya produk yang di beli berdasarkan product line

- 3) Menampilkan bar perbandingan pendapatan dari membership berdasarkan member atau normal yang akan memanggil index Customer Type.

Source code:

```
## bar ##
Membership = df.groupby('Customer type').sum()
membership = [Customer_type for Customer_type, df in df.groupby('Customer type')]
plt.figure(figsize=(10, 5))
plt.bar(membership, Membership['Total'])
plt.title('Perbandingan Pendapatan dari Membership', fontsize=15)
plt.xticks(membership)
plt.ylabel('Total dalam USD ($)', fontsize=12)
plt.xlabel('Jenis Pelanggan', fontsize=12)
plt.grid()
plt.show()
```

Gambar 21. Menampilkan bar diagram perbandingan pendapatan dari tipe membership

Output:



Gambar 22. Output bar diagram perbandingan pendapatan dari tipe membership

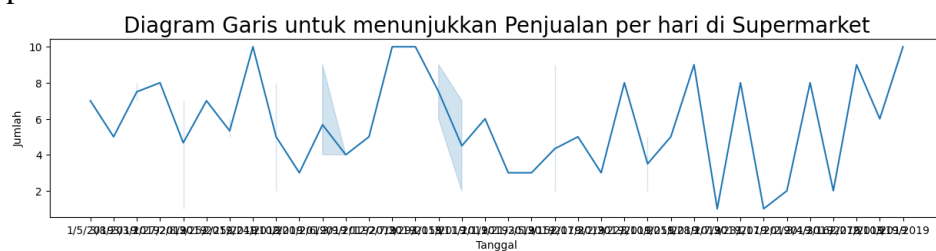
- 4) Menampilkan diagram garis untuk menunjukkan penjualan per hari yang akan memanggil index Date dan Quantity.

Source code:

```
## Garis ##
plt.figure(figsize=(15, 3))
sns.lineplot(x='Date', y='Quantity', data=df.iloc[:n])
plt.title('Diagram Garis untuk menunjukkan Penjualan per hari di Supermarket', fontdict={'fontsize': 20})
plt.xlabel('Tanggal')
plt.ylabel('Jumlah')
plt.show()
```

Gambar 23. Menampilkan diagram garis penjualan per hari

Output:



Gambar 24. Output diagram garis penjualan per hari

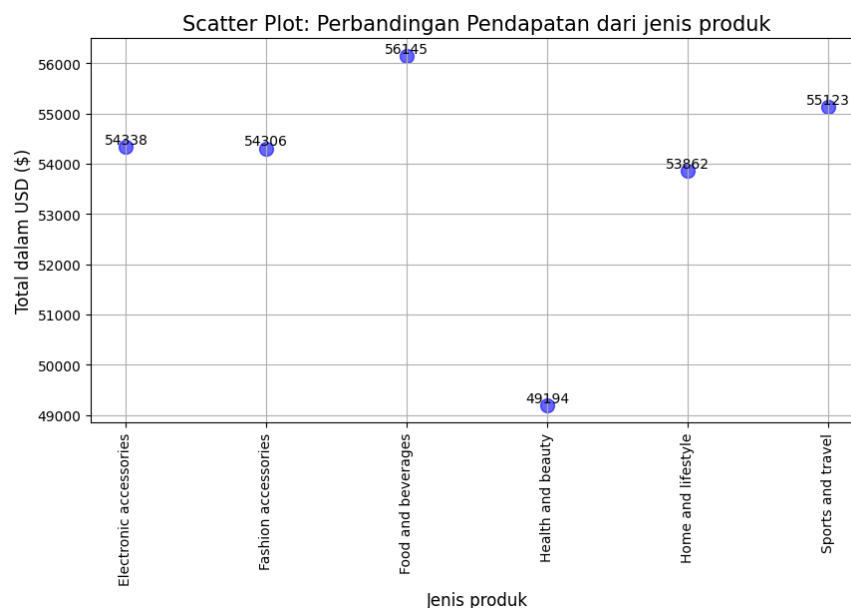
- 5) Menampilkan Scatter plot perbandingan pendapatan dari jenis produk yang akan memanggil index Product line dan Total.

Source code:

```
## scatter Plot ##
jenis_produk = df.groupby('Product line').sum()
Jenis_produk = [Customer_type for Customer_type, df in df.groupby('Product line')]
plt.figure(figsize=(10, 5))
plt.scatter(Jenis_produk[:n], jenis_produk['Total'][:n], s=100, c='b', alpha=0.6)
plt.title('Scatter Plot: Perbandingan Pendapatan dari jenis produk', fontsize=15)
plt.xticks(rotation='vertical')
plt.xlabel('Jenis produk', fontsize=12)
plt.ylabel('Total dalam USD ($)', fontsize=12)
plt.grid()
# Menambahkan label angka
for i, j in enumerate(jenis_produk['Total'][:n]):
    plt.text(i, j, f'{j:.0f}', ha='center', va='bottom', fontsize=10)
plt.show()
```

Gambar 25. Menampilkan Scatter plot pendapatan dari jenis produk

Output:



Gambar 26. Output Scatter plot pendapatan dari jenis produk

Kesimpulan

Analisis dan Visualisasi Data Laporan Penjualan Toko Online melalui pendekatan Data Science menggunakan Python adalah proses yang sangat bermanfaat dalam dunia perdagangan daring. Dalam era digital ini, data menjadi harta karun yang memberikan wawasan berharga kepada pemilik toko online dan pengambil keputusan.

Analisis dan Visualisasi Data Laporan Penjualan Toko Online melalui pendekatan Data Science menggunakan Python adalah proses yang sangat bermanfaat dalam dunia

perdagangan daring. Dalam era digital ini, data menjadi harta karun yang memberikan wawasan berharga kepada pemilik toko online dan pengambil keputusan.

Daftar Pustaka

- [1] Written by Amira K, “Data Science: Pengertian, Alat yang Digunakan, dan Penerapannya,” https://www.gramedia.com/literasi/data-science-adalah/#Pengertian_Data_Science. https://www.gramedia.com/literasi/data-science-adalah/#Pengertian_Data_Science
- [2] “Apa itu Visualisasi Data?,” <https://aws.amazon.com>. <https://aws.amazon.com/id/what-is/data-visualization/>
-