

## **LAPORAN TUGAS BESAR**

### **“PEMBANGUNAN VISUALISASI DATA MENGGUNAKAN BAHASA PEMOGRAMAN PYTHON”**

Disusun Untuk Memenuhi Tugas Mata Kuliah Analisis dan Visualisasi Data

**Dosen Pengampu : Ricky Akbar, M.Kom dan Rahmantika Pratama Santi, M.T**



Oleh :

#### **Kelompok 1**

1. Nadini Annisa Byant 2111522021
2. Ahmad Rifki Farid 2111523009
3. Mhd. Faiz Yunus 2111523010
4. Syafira Putri Zahra 2111523013

**DEPARTEMEN SISTEM INFORMASI  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
2023**

## **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua. Shalawat serta salam semoga tercurah kepada Rasulullah SAW beserta keluarganya. Tidak lupa penulis mengucapkan terima kasih kepada bapak Ricky Akbar, M.Kom dan ibu Rahmantika Pratama Santi, M.T selaku dosen pengampu mata kuliah Analisis dan Visualisasi Data. Penulis juga mengucapkan terima kasih kepada teman-teman yang telah memberi kontribusi baik secara langsung maupun tidak langsung dalam pembuatan laporan ini.

Dalam penyusunan makalah ini, penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kesempurnaan karena pengalaman dan pengetahuan penulis yang terbatas. Oleh karena itu, kritik dan saran dari semua pihak sangat penulis harapkan demi terciptanya makalah yang lebih baik lagi untuk masa mendatang.

Padang, 13 May 2023

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>INFORMASI YANG DITAMPILKAN.....</b>	<b>1</b>
A...Penjelasan Dataset.....	1
B...Informasi yang Ditampilkan.....	1
<b>IMPLEMENTASI VISUALISASI DATA.....</b>	<b>4</b>

## INFORMASI YANG DITAMPILKAN

### A. Penjelasan Dataset

Dataset ini berisi data penjualan supermarket dengan beberapa kolom yang mencakup informasi tentang transaksi, produk, pelanggan, dan lainnya. Berikut adalah penjelasan kolom-kolom yang ada dalam dataset tersebut:

- Branch: Menyatakan cabang atau outlet supermarket (misalnya, Branch A, Branch B, Branch C).
- City: Menyatakan kota tempat ber lokasinya cabang supermarket.
- Customer type: Menyatakan jenis pelanggan (misalnya, Member, Normal).
- Gender: Menyatakan jenis kelamin pelanggan (misalnya, Male, Female).
- Product line: Menyatakan kategori atau jenis produk yang dibeli oleh pelanggan.
- Unit price: Menyatakan harga satuan produk.
- Quantity: Menyatakan jumlah produk yang dibeli.
- Tax: Menyatakan jumlah pajak yang dikenakan pada transaksi.
- Total: Menyatakan total biaya pembelian.
- Date: Menyatakan tanggal transaksi.
- Time: Menyatakan waktu transaksi.
- Payment: Menyatakan metode pembayaran (misalnya, Cash, Credit Card, E-Wallet).
- COGS (Cost of Goods Sold): Menyatakan biaya barang yang terjual.
- Gross margin percentage: Menyatakan persentase margin keuntungan kotor.
- Gross income: Menyatakan total keuntungan kotor.

### B. Informasi yang Akan Ditampilkan

1. Jumlah Transaksi yang Dilakukan di Tiap kota
2. Jenis Produk yang Paling Banyak Dibeli oleh Pelanggan
3. Jenis Produk yang Paling Sedikit Dibeli oleh Pelanggan
4. Rating Pelanggan Terhadap Transaksi pada Tiap Jenis Produk
5. Rating Pelanggan Terhadap Transaksi pada Tiap Cabang
6. Persentase Margin Keuntungan Kotor pada Tiap Jenis Produk

7. Persentase Margin Keuntungan Kotor pada Tiap Cabang
8. Jumlah Produk yang Dibeli oleh Jenis Kelamin Pelanggan untuk setiap Kategori Produk
9. Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Cabang
10. Persentase Tipe Customer secara Keseluruhan dan dari Tiap Cabang/Kota
11. Persebaran Pembelian Tiap Bulan per Cabang
12. Rata-rata Rating Pelanggan untuk Setiap Kategori Produk
13. Jumlah Pelanggan berdasarkan Jenis kelamin pada Setiap Kota
14. Persentase Penjualan setiap Kategori Produk pada Setiap cabang/Kota
15. Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Jenis Kelamin
16. Persentase Produk yang dibeli oleh Pelanggan Member
17. Persentase Produk yang dibeli oleh Pelanggan Non-Member
18. Jumlah Transaksi pada Setiap Jam untuk setiap Cabang/Kota
19. Total Biaya Pembelian pada setiap Kategori Produk untuk setiap Cabang/Kota
20. Persentase Rating Pelanggan yang Memberikan Rating 7 ke atas untuk setiap Kategori Produk
21. Jumlah Transaksi untuk Setiap Kategori Produk untuk Setiap Cabang/Kota
22. Total Biaya Pembelian pada Setiap Jenis Produk untuk setiap Jenis
23. Total Gross Income per Jenis kelamin
24. Persebaran Payment per Jenis Kelamin
25. Total Product Line per Jenis Kelamin
26. Total dari Total per Jenis Kelamin
27. Total Quantity per Jenis Kelamin
28. Total Tax per Cabang
29. Total Tax per City
30. Total Tax per Gender
31. Total Tax per Product Line
32. Total Gross Income Terbanyak Tiap Branch
33. Total Gross Margin Tiap Branch
34. Total pembelian Tiap Branch
35. Total Tax per Branch
36. Total Tipe Customer Tiap Branch
37. Total Mean Rating per Branch

- 38. Total Payment per Branch
- 39. Total Produk Tiap Branch
- 40. Total Kota Tiap Branch

## IMPLEMENTASI VISUALISASI DATA

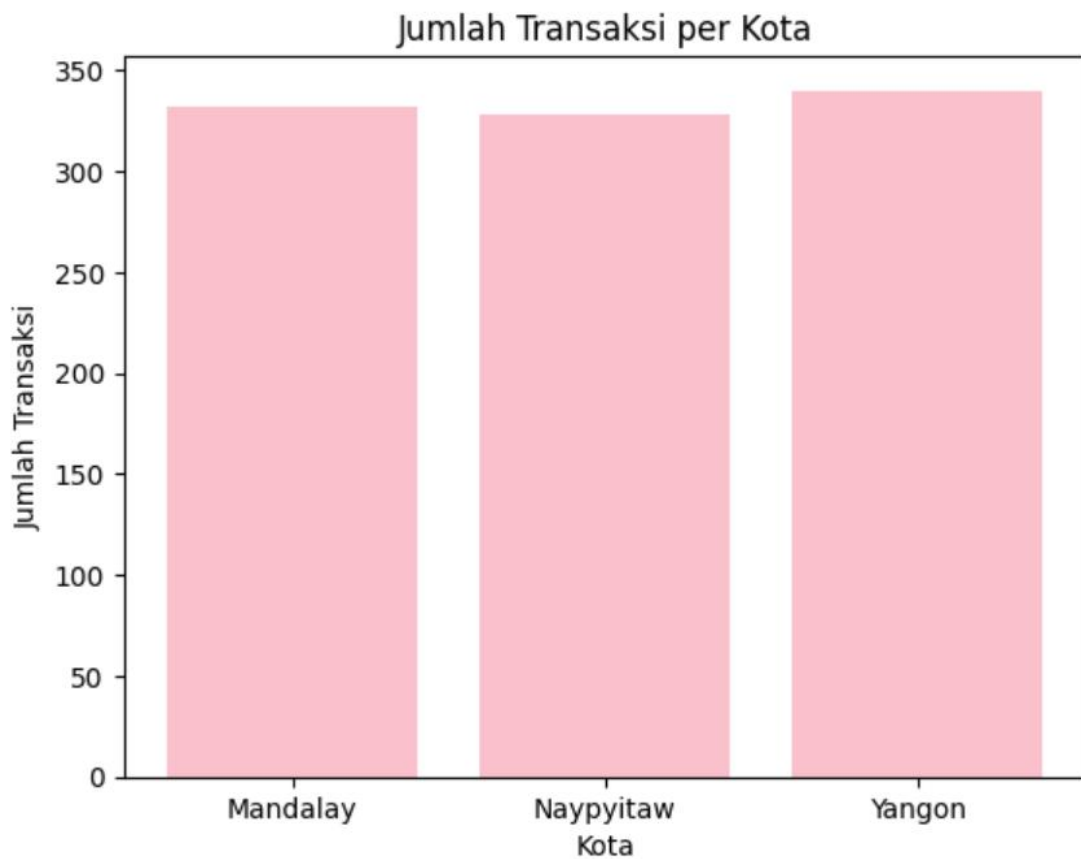
1. Jumlah transaksi yang dilakukan di tiap kota

```
# kelompokkan data berdasarkan city lalu jumlahkan jumlah transaksi nya
transactions_per_city = smoker.groupby('City')['Invoice ID'].count()

# buat chart -> x = kota transaksi, y = jumlah transaksi
plt.bar(transactions_per_city.index, transactions_per_city.values, color='pink')

# label chart
plt.title('Jumlah Transaksi per Kota')
plt.xlabel('Kota')
plt.ylabel('Jumlah Transaksi')

# tampilkan chart
plt.show()
```



Penjelasan:

Pada bagian ini akan ditampilkan jumlah transaksi perdata kotanya, dimana dataset terdapat 3 kota yaitu mandalay, naypyitaw, dan yangon. Pada kode tersebut akan dikelompokkan data transaksi berdasarkan invoice id dan dijumlahkan banyak transaksinya. Lalu akan ditampilkan dengan grafik chart, dimana sumbu x dijadikan sebagai kota transaksi, dan sumbu y dijadikan sebagai jumlah transaksi. Sehingga dihasilkan hasil visualisasi data seperti gambar diatas yang berarti kota yangon memegang jumlah transaksi terbanyak, dimana ke 3 kota tersebut memiliki jumlah transaksi yaitu 300-350, yang diperkirakan kota mandalay 330 transaksi, kota naypyitaw 320 transaksi, dan yangon 340 transaksi.

## 2. Jenis produk yang paling banyak dibeli oleh pelanggan

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

# membaca file csv
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')
# menghitung total jumlah produk berdasarkan jenis produk
product_counts = df['Product line'].value_counts()

# mengambil jenis produk yang paling banyak dibeli
most_bought_product = product_counts.index[0]

# menampilkan jenis produk yang paling banyak dibeli dalam bentuk pie
chart
plt.pie(product_counts, labels=product_counts.index, autopct='%1.1f%%')
plt.title(f'Most Bought Product: {most_bought_product}')
plt.axis('equal')
plt.show()
```

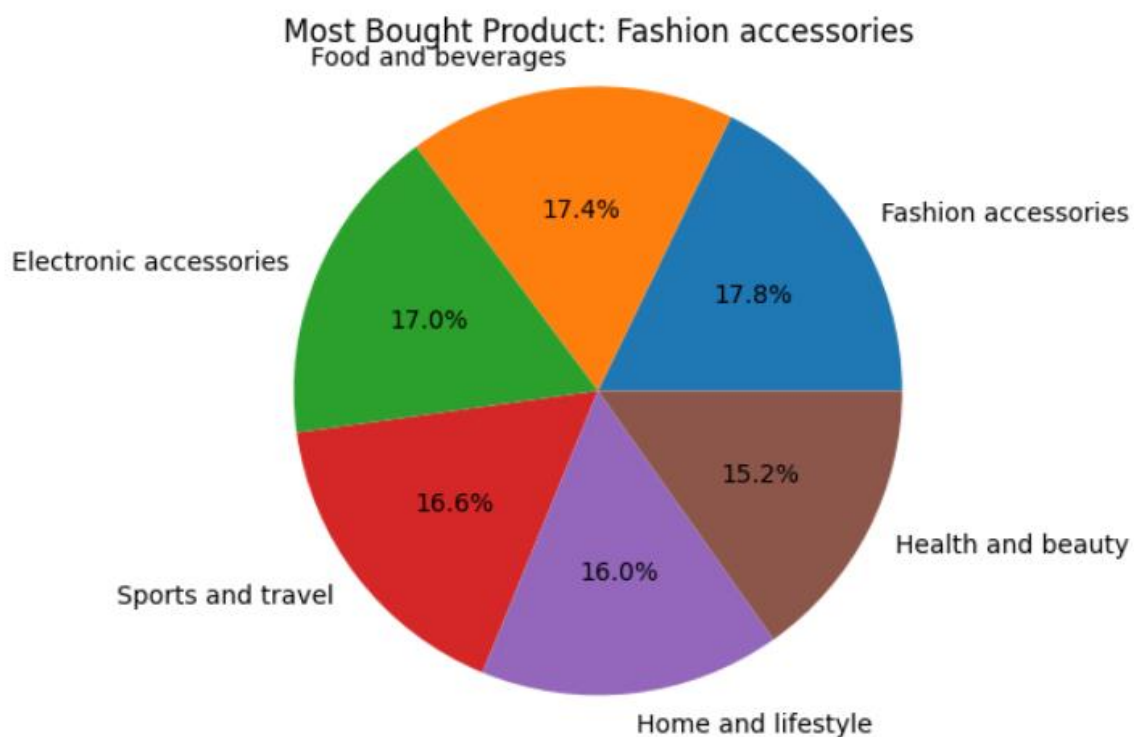
Penjelasan :

Pada bagian ini akan ditampilkan jenis produk yang paling banyak dibeli oleh pelanggan, dimana pada dataset terdapat kolom data product line, yang berisikan beberapa jenis produk yaitu electronic accessories, food and beverages, sports and travel, home and lifestyle, fashion accessories, health and beauty.

Kemudian menghitung total jumlah produk berdasarkan jenis produk. Baris kode `product_counts = df['Product line'].value_counts()` menghitung jumlah kemunculan setiap jenis produk dalam kolom 'Product line' dan menyimpannya dalam variabel `product_counts`. lalu, mengambil jenis produk yang paling banyak dibeli. Baris kode `most_bought_product = product_counts.index[0]` mengambil indeks pertama dari `product_counts`, yang akan menjadi jenis produk yang paling banyak



dibeli. Terakhir, membuat pie chart dari data `product_counts`. Baris kode `plt.pie(product_counts, labels=product_counts.index, autopct='%1.1f%%')` membuat pie chart dengan menggunakan data `product_counts`. Kemudian, `plt.title(f'Most Bought Product: {most_bought_product}')` menambahkan judul pada pie chart, yang menampilkan jenis produk yang paling banyak dibeli. `plt.axis('equal')` digunakan untuk memastikan pie chart memiliki proporsi yang benar. Akhirnya, `plt.show()` digunakan untuk menampilkan pie chart tersebut seperti pada gambar berikut. Persentase penjualan barang yang paling banyak adalah pada kategori *fashion accessories* dengan persentase sebesar 17.8%.



### 3. Jenis produk yang paling sedikit dibeli oleh pelanggan

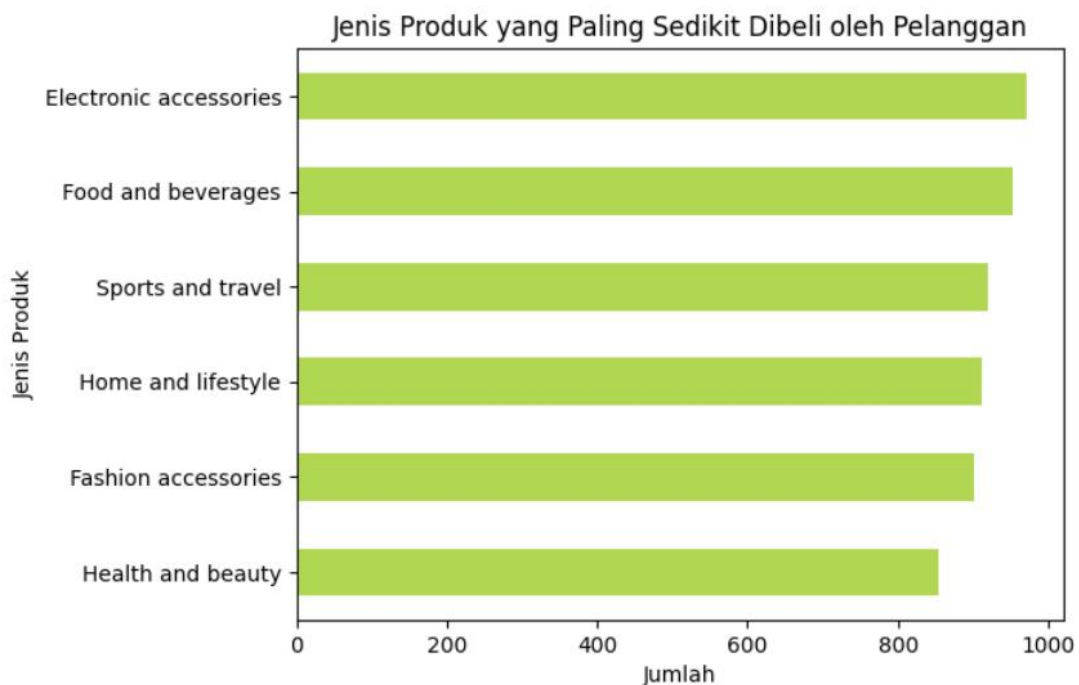
```
# mengelompokkan data berdasarkan kolom product line lalu menjumlahkannya
# berdasarkan kolom quantity
product_qty = smoker.groupby('Product line')['Quantity'].sum()

# ascending jumlah produk
product_qty = product_qty.sort_values(ascending=True)

# membuat car chart menjadi horizontal
product_qty.plot(kind='barh', color='yellowgreen', alpha=0.8)
```

```
# mengisi judul dan label bar chart
plt.title("Jenis Produk yang Paling Sedikit Dibeli oleh Pelanggan")
plt.xlabel("Jumlah")
plt.ylabel("Jenis Produk")

# menampilkan bar chart
plt.show()
```



Penjelasan:

Pada bagian ini akan ditampilkan jenis produk yang paling sedikit dibeli oleh pelanggan, dimana pada dataset terdapat kolom data product line, yang berisikan beberapa jenis produk yaitu electronic accessories, food and beverages, sports and travel, home and lifestyle, fashion accessories, health and beauty. Pada kode tersebut akan dikelompokkan data berdasarkan jenis produk (product line) dan banyak produk yang dibeli dan nantinya akan dijumlahkan. data yang dikelompokkan itu disimpan pada sebuah variabel yaitu product\_qty. lalu hasil dari penjumlahan tersebut akan ditampilkan secara bar chart horizontal.

Sehingga dihasilkan visualisasi data seperti gambar diatas yang berarti jenis produk yang paling sedikit dibeli oleh pelanggan adalah health dan beauty, dan juga dapat dilihat urutan produk dari banyak dibeli hingga sedikit yaitu adalah electronic accessories, food and beverages, sports and travel, home and lifestyle, fashion accessories, dan yang paling sedikit health and beauty.

#### 4. Rating Pelanggan terhadap Transaksi pada Tiap Jenis Produk

```
# 4. Total biaya pembelian pada tiap metode pembayaran
import pandas as pd
import matplotlib.pyplot as plt

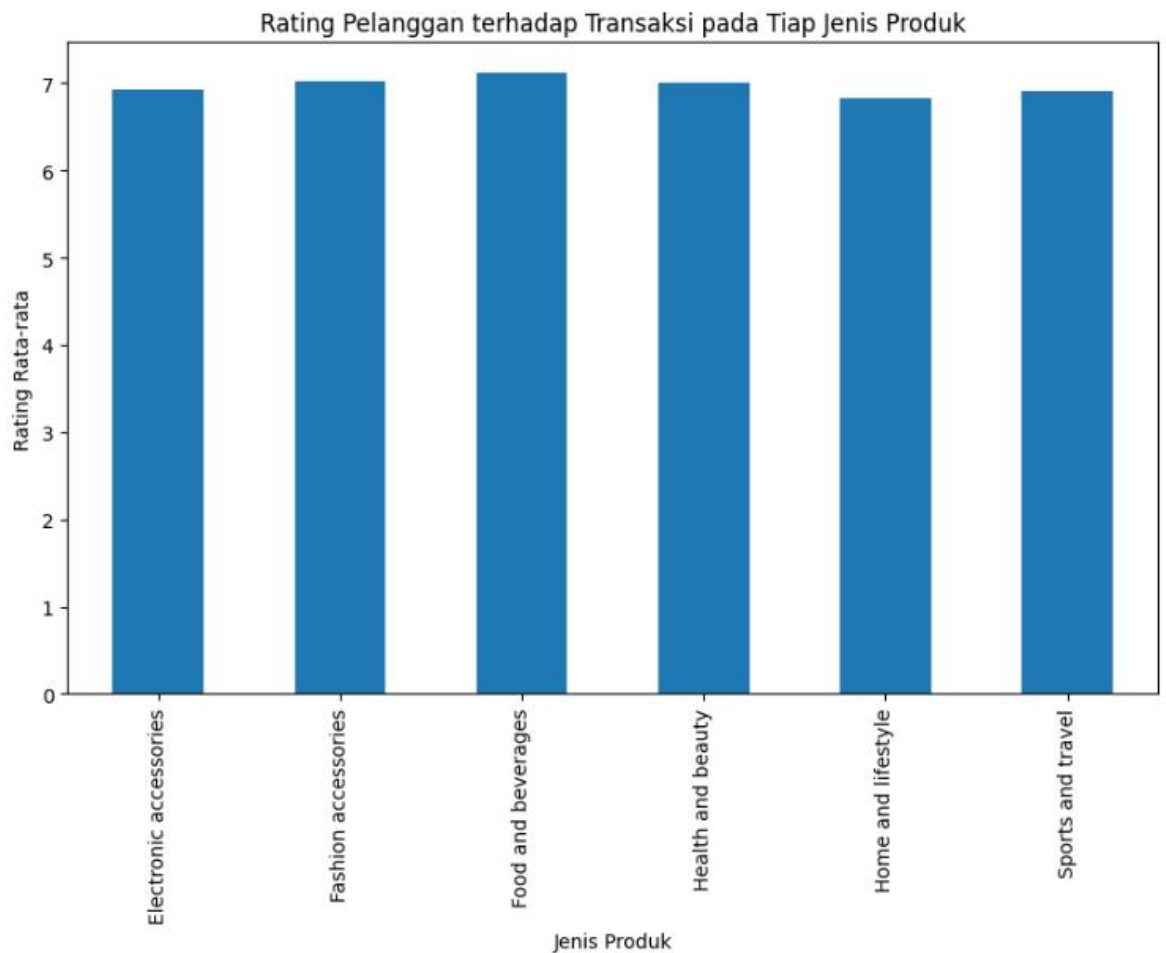
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung rata-rata rating pelanggan untuk setiap jenis produk
rating_per_produk = df.groupby('Product line')['Rating'].mean()

# Membuat bar chart
fig, ax = plt.subplots(figsize=(10, 6))
rating_per_produk.plot(kind='bar', ax=ax)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Jenis Produk')
plt.ylabel('Rating Rata-rata')
plt.title('Rating Pelanggan terhadap Transaksi pada Tiap Jenis Produk')

# Menampilkan bar chart
plt.show()
```



Penjelasan:

Pada bagian ini akan ditampilkan jenis produk yang paling sedikit dibeli oleh pelanggan, dimana pada dataset terdapat kolom data product line, yang berisikan beberapa jenis produk yaitu electronic accessories, food and beverages, sports and travel, home and lifestyle, fashion accessories, health and beauty. Pada kode tersebut akan dikelompokkan data berdasarkan jenis produk (product line) dan banyak produk yang dibeli dan nantinya akan dijumlahkan. data yang dikelompokkan itu disimpan pada sebuah variabel yaitu product\_qty. lalu hasil dari penjumlahan tersebut akan ditampilkan secara bar chart horizontal.

Sehingga dihasilkan visualisasi data seperti gambar diatas yang berarti jenis produk yang paling sedikit dibeli oleh pelanggan adalah health dan beauty, dan juga dapat dilihat urutan produk dari banyak dibeli hingga sedikit yaitu adalah electronic accessories, food and beverages, sports and travel, home and lifestyle, fashion accessories, dan yang paling sedikit health and beauty.

5. Rating pelanggan terhadap transaksi pada tiap cabang

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

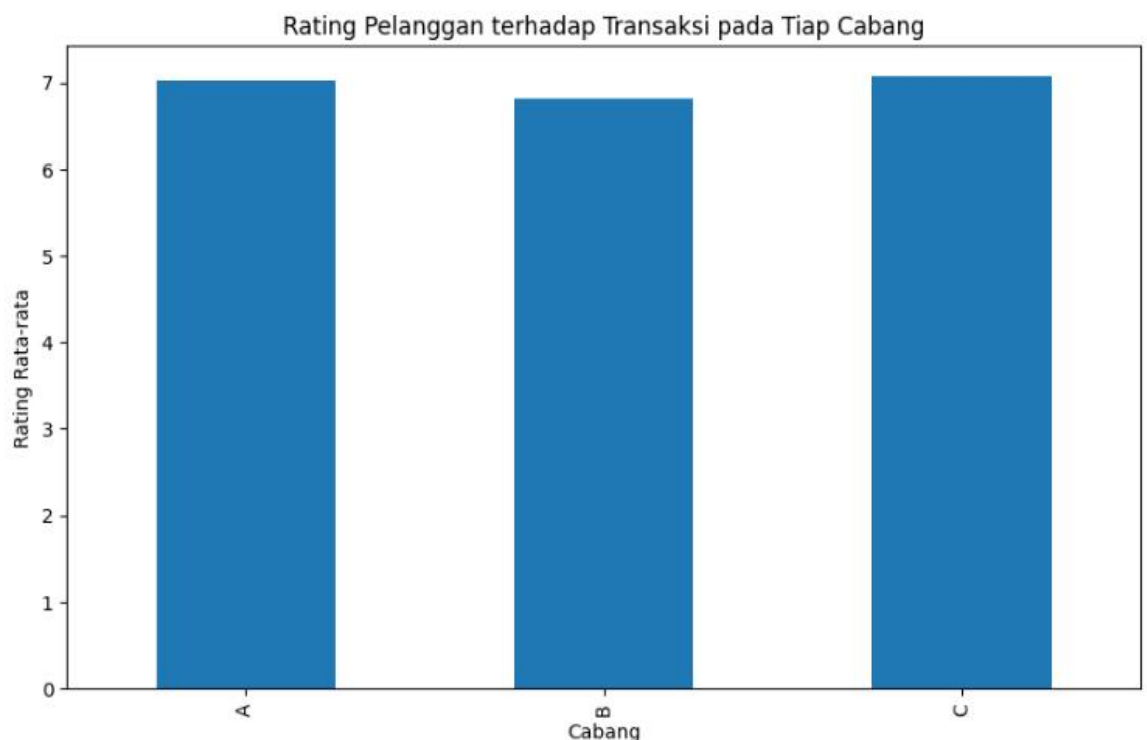
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung rata-rata rating pelanggan untuk setiap cabang
rating_per_cabang = df.groupby('Branch')['Rating'].mean()

# Membuat bar chart
fig, ax = plt.subplots(figsize=(10, 6))
rating_per_cabang.plot(kind='bar', ax=ax)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Cabang')
plt.ylabel('Rating Rata-rata')
plt.title('Rating Pelanggan terhadap Transaksi pada Tiap Cabang')

# Menampilkan bar chart
plt.show()
```



Penjelasan:

Codingan tersebut menghitung rata-rata rating pelanggan untuk setiap cabang dalam DataFrame 'df'. Berikut adalah penjelasan langkah-langkahnya:

`df.groupby('Branch')`: Perintah ini mengelompokkan baris-baris dalam DataFrame 'df' berdasarkan nilai pada kolom 'Branch'. Ini berarti baris-baris yang memiliki nilai 'Branch' yang sama akan dikelompokkan bersama-sama.

Setelah melakukan pengelompokan, kita memilih kolom 'Rating' dari setiap kelompok. Ini berarti kita akan memperoleh semua nilai rating yang terkait dengan setiap cabang. Metode `mean()` diterapkan pada kolom 'Rating' yang dipilih. Ini akan menghitung rata-rata dari semua nilai rating yang terkait dengan setiap cabang.

Hasilnya adalah sebuah Series yang berisi nilai rata-rata rating pelanggan untuk setiap cabang. Indeks dari Series ini akan terdiri dari nilai-nilai unik dari kolom 'Branch'. Anda dapat menyimpan hasilnya ke dalam variabel 'rating\_per\_cabang' untuk digunakan selanjutnya dalam analisis atau tampilan hasil dengan menggunakan `print(rating_per_cabang)`.

#### 6. Persentase margin keuntungan kotor pada tiap jenis produk

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

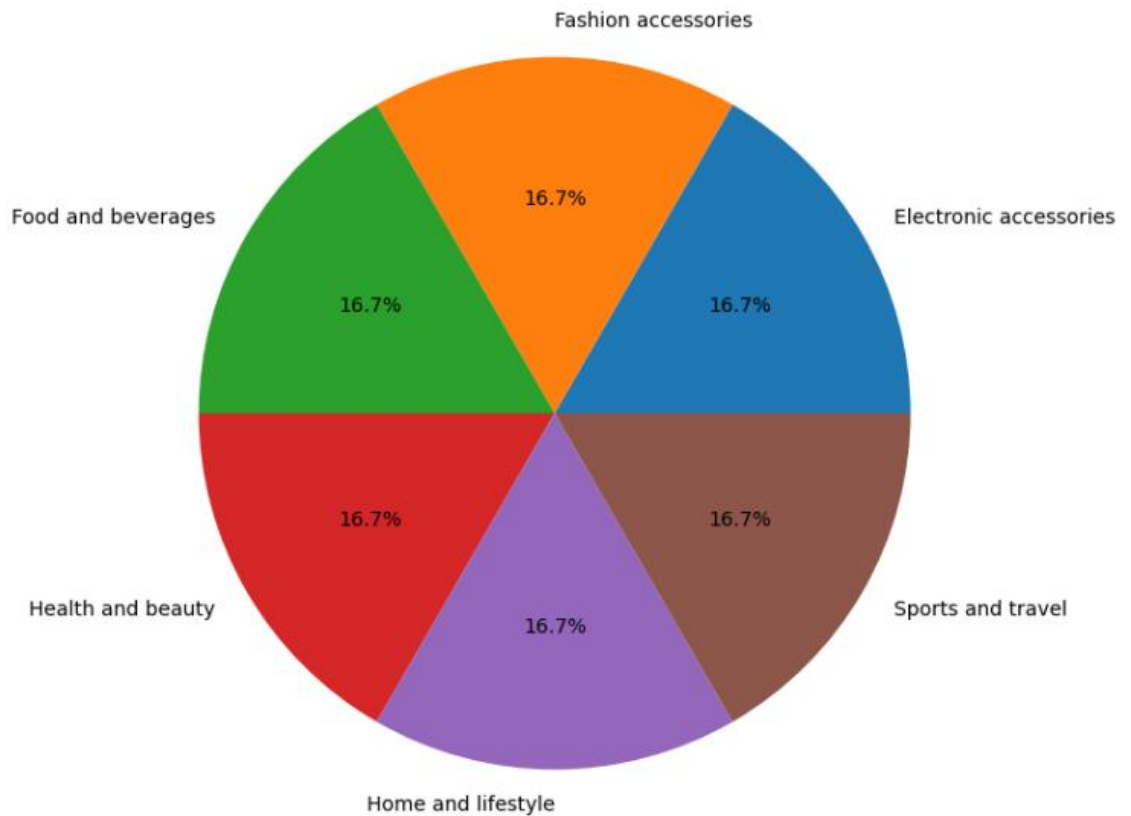
# Menghitung total margin keuntungan kotor pada tiap jenis produk
margin_per_produk = df.groupby('Product line')['gross margin percentage'].mean()

# Membuat pie chart
fig, ax = plt.subplots(figsize=(8, 8))
ax.pie(margin_per_produk, labels=margin_per_produk.index, autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Persentase Margin Keuntungan Kotor pada Tiap Jenis Produk')

# Menampilkan pie chart
plt.show()
```

Persentase Margin Keuntungan Kotor pada Tiap Jenis Produk



Penjelasan:

Dengan menggunakan metode `'groupby'` pada kolom `'Product line'`, kita mengelompokkan data berdasarkan jenis produk dan kemudian menghitung rata-rata margin keuntungan kotor untuk setiap kelompok. Hasil perhitungan ini disimpan dalam variabel `'margin_per_produk'`.

Selanjutnya, kita menggunakan metode `'pie'` pada objek sumbu `'ax'` untuk membuat pie chart. Parameter pertama adalah data yang akan diplot, yaitu `'margin_per_produk'`. Parameter `'labels'` digunakan untuk menentukan label pada setiap sektor pie chart berdasarkan indeks `'margin_per_produk.index'`. Parameter `'autopct'` digunakan untuk menampilkan persentase pada setiap sektor dengan format `'%1.1f%%'`.

#### 7. Persentase margin keuntungan kotor pada tiap cabang

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')
```

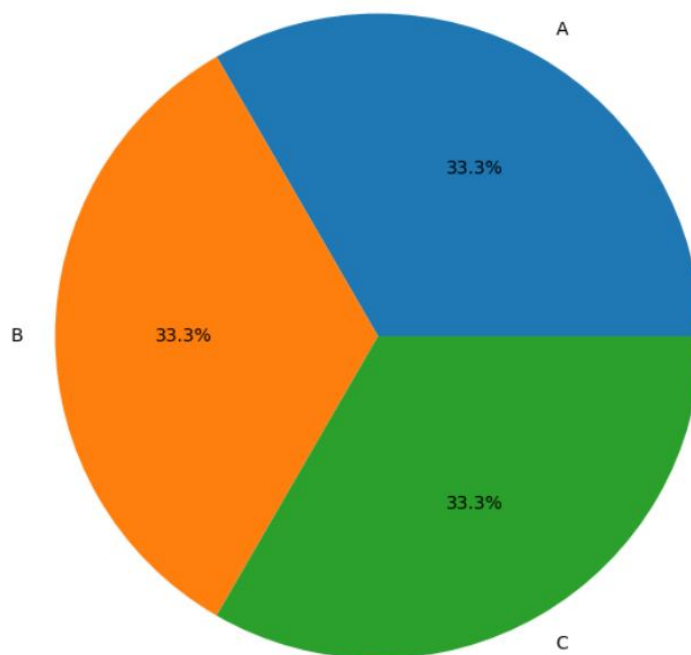
```
# Menghitung total margin keuntungan kotor pada tiap cabang
margin_per_cabang = df.groupby('Branch')['gross margin percentage'].mean()

# Membuat pie chart
fig, ax = plt.subplots(figsize=(8, 8))
ax.pie(margin_per_cabang, labels=margin_per_cabang.index, autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Persentase Margin Keuntungan Kotor pada Tiap Cabang')

# Menampilkan pie chart
plt.show()
```

Persentase Margin Keuntungan Kotor pada Tiap Cabang



Penjelasan:

Dengan menggunakan metode `groupby` pada kolom 'Branch', kita mengelompokkan data berdasarkan cabang dan kemudian menghitung rata-rata margin keuntungan kotor untuk setiap cabang. Hasil perhitungan ini disimpan dalam variabel `margin\_per\_cabang`.

Selanjutnya, kita menggunakan metode `pie` pada objek sumbu `ax` untuk membuat pie chart. Parameter pertama adalah data yang akan diplot, yaitu `margin\_per\_cabang`. Parameter `labels` digunakan untuk menentukan label pada setiap sektor pie chart berdasarkan indeks `margin\_per\_cabang.index`. Parameter `autopct` digunakan untuk menampilkan persentase pada setiap sektor dengan format `%1.1f%%`.



8. Jumlah produk yang dibeli oleh jenis kelamin pelanggan untuk setiap kategori produk

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

# membaca file csv
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# menghitung jumlah produk yang dibeli berdasarkan jenis kelamin dan kategori produk
produk_per_gender = df.groupby(['Gender', 'Product line']).size().unstack()

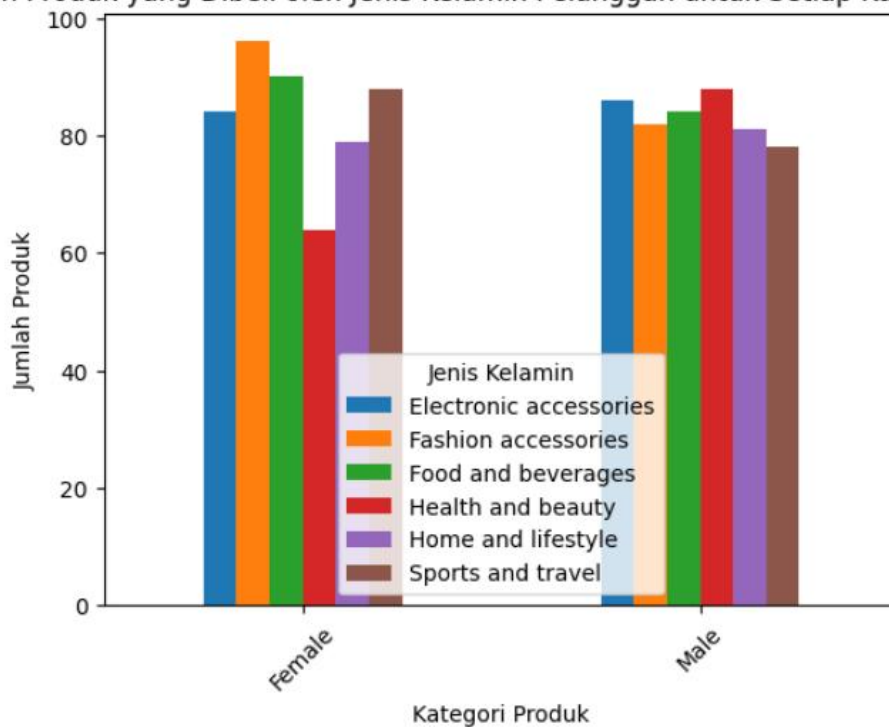
# menampilkan column chart
produk_per_gender.plot(kind='bar')
plt.xlabel('Kategori Produk')
plt.ylabel('Jumlah Produk')
plt.title('Jumlah Produk yang Dibeli oleh Jenis Kelamin Pelanggan untuk Setiap Kategori Produk')
plt.legend(title='Jenis Kelamin')
plt.xticks(rotation=45)
plt.show()
```

Penjelasan :

Pada bagian ini ditampilkan Jumlah Produk yang Dibeli oleh Jenis Kelamin Pelanggan untuk Setiap Kategori Produk dimana pada dataset terdapat kolom *gender* dan *product line*. Baris kode `produk_per_gender = df.groupby(['Gender', 'Product line']).size().unstack()` mengelompokkan data berdasarkan kolom 'Gender' dan 'Product line', kemudian menghitung jumlah produk untuk setiap kombinasi jenis kelamin dan kategori produk. Hasilnya disusun dalam bentuk tabel dengan indeks 'Gender' dan kolom-kolom adalah kategori produk.

Kemudian, membuat column chart dari data `produk_per_gender`. Baris kode `produk_per_gender.plot(kind='bar')` membuat column chart dengan menggunakan data `produk_per_gender`. Kemudian, `plt.xlabel('Kategori Produk')` digunakan untuk memberikan label pada sumbu x (kategori produk), `plt.ylabel('Jumlah Produk')` digunakan untuk memberikan label pada sumbu y (jumlah produk), `plt.title('Jumlah Produk yang Dibeli oleh Jenis Kelamin Pelanggan untuk Setiap Kategori Produk')` digunakan untuk memberikan judul pada chart, `plt.legend(title='Jenis Kelamin')` digunakan untuk menampilkan legenda dengan judul 'Jenis Kelamin', `plt.xticks(rotation=45)` digunakan untuk memutar label sumbu x sebesar 45 derajat agar lebih mudah terbaca. Terakhir, `plt.show()` digunakan untuk menampilkan column chart tersebut. Sehingga didapatkan hasil visualisasi menggunakan *column chart* sebagai berikut :

Jumlah Produk yang Dibeli oleh Jenis Kelamin Pelanggan untuk Setiap Kategori Produk



9. Total biaya pembelian pada tiap jenis produk untuk setiap cabang

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

smoker = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total biaya pembelian pada tiap jenis produk untuk setiap
cabang/kota
total_biaya = smoker.groupby(['Branch', 'City', 'Product
line'])['cogs'].sum().unstack().reset_index()

# Menyiapkan data untuk pie chart
cabang_kota = total_biaya[['Branch', 'City']].apply(lambda x: f'{x[0]} - {x[1]}',
axis=1)
biaya = total_biaya.values[:, 3:]

# Membuat pie chart
fig, ax = plt.subplots(figsize=(10, 6))

# Menghitung total biaya pembelian per cabang/kota
total_biaya_cabang = biaya.sum(axis=1)

# Menghitung persentase biaya pembelian untuk setiap cabang/kota
persentase_biaya = biaya / total_biaya_cabang[:, None] * 100

# Menampilkan pie chart
```

```

colors = plt.cm.Set3(range(len(cabang_kota)))
ax.pie(total_biaya_cabang, labels=cabang_kota, autopct='%1.1f%%', colors=colors)

# Menentukan judul chart
plt.title('Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Cabang/Kota')

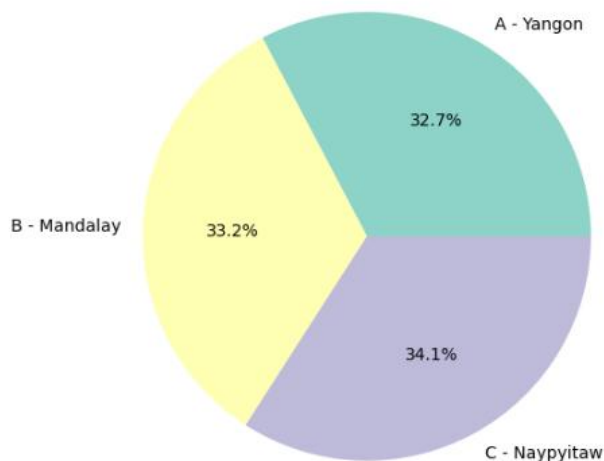
# Menampilkan pie chart
plt.show()

```

Penjelasan :

Pada bagian ini akan ditampilkan total biaya pembelian pada tiap jenis produk untuk perdata kotanya, dimana dataset terdapat 3 kota yaitu Mandalay, Naypyitaw, dan Yangon. dimana disini data Branch, City dan Product line untuk menghitung total biaya pembelian dengan codingan program ini `total_biaya = smoker.groupby(['Branch', 'City', 'Product line'])['cogs'].sum().unstack().reset_index()`, setelah itu Menghitung total biaya pembelian per cabang/kota dengan codingan berikut : `total_biaya_cabang = biaya.sum(axis=1)`, dan terakhir kita akan menentukan persentase biaya pembelian untuk setiap cabang / kota dengan codingan program `persentase_biaya = biaya / total_biaya_cabang[:, None] * 100`. Maka dapat dilihat visualisasi datanya sebagai berikut :

Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Cabang/Kota



#### 10. Persentase tipe customer secara keseluruhan dan dari tiap cabang/kota

```
import pandas as pd
```

```
import matplotlib.pyplot as plt

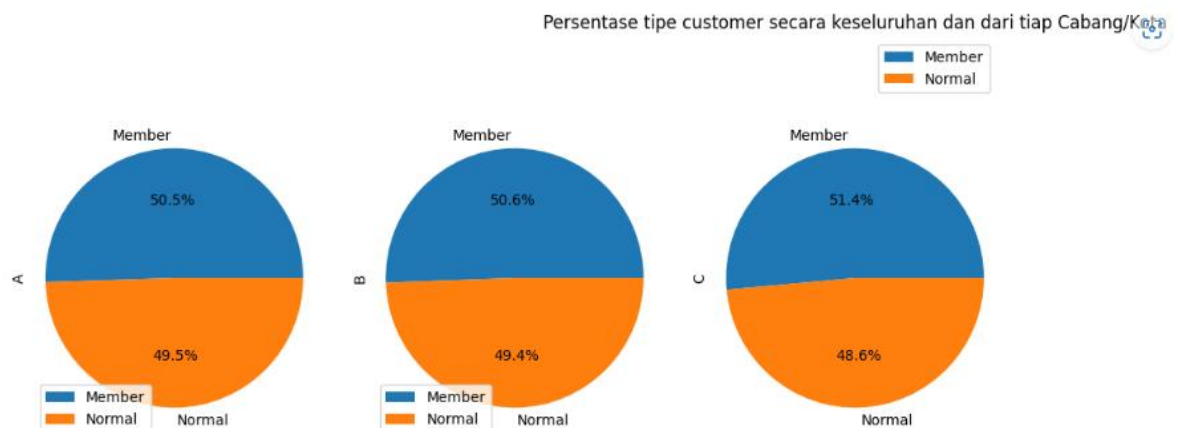
# menghitung total penjualan tipe customer secara keseluruhan pada setiap
cabang/kota
penjualan_per_cabang = smoker.groupby(['Branch', 'Customer
type'])['Total'].sum().unstack()

# menampilkan pie chart
penjualan_per_cabang.T.plot(kind='pie', subplots=True, figsize=(12, 6),
autopct='%1.1f%%')
plt.axis('equal')
plt.title('Persentase tipe customer secara keseluruhan dan dari tiap Cabang/Kota')
plt.legend(loc='upper right')
plt.show()
```

Penjelasan :

Pada bagian ini akan ditampilkan jumlah transaksi perdata kotanya, dimana dataset terdapat 3 kota yaitu mandalay, naypyitaw, dan yangon. Pada kode tersebut akan dikelompokkan data transaksi berdasarkan customer type dan dijumlahkan banyak transaksinya.

Lalu akan ditampilkan dengan grafik chart, dimana sumbu x dijadikan sebagai kota transaksi, dan sumbu y dijadikan sebagai jumlah transaksi. Sehingga dihasilkan hasil visualisasi data seperti gambar diatas yang berarti kota yangon memegang jumlah transaksi terbanyak, dimana ke 3 kota tersebut memiliki jumlah transaksi yaitu 300-350, yang diperkirakan kota mandalay 330 transaksi, kota naypyitaw 320 transaksi, dan yangon 340 transaksi.



## 11. Persebaran pembelian tiap bulan per cabang

```

import pandas as pd
import matplotlib.pyplot as plt

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Mengubah kolom 'Date' menjadi tipe data datetime
df['Date'] = pd.to_datetime(df['Date'])

# Menambahkan kolom 'Month' yang berisi informasi bulan dari kolom 'Date'
df['Month'] = df['Date'].dt.month

# Menghitung total pembelian tiap bulan percabang
pembelian_per_bulan_percabang = df.groupby(['Month',
'Branch'])['Total'].sum().unstack()

# Membuat line chart persebaran pembelian tiap bulan percabang
fig, ax = plt.subplots(figsize=(10, 6))

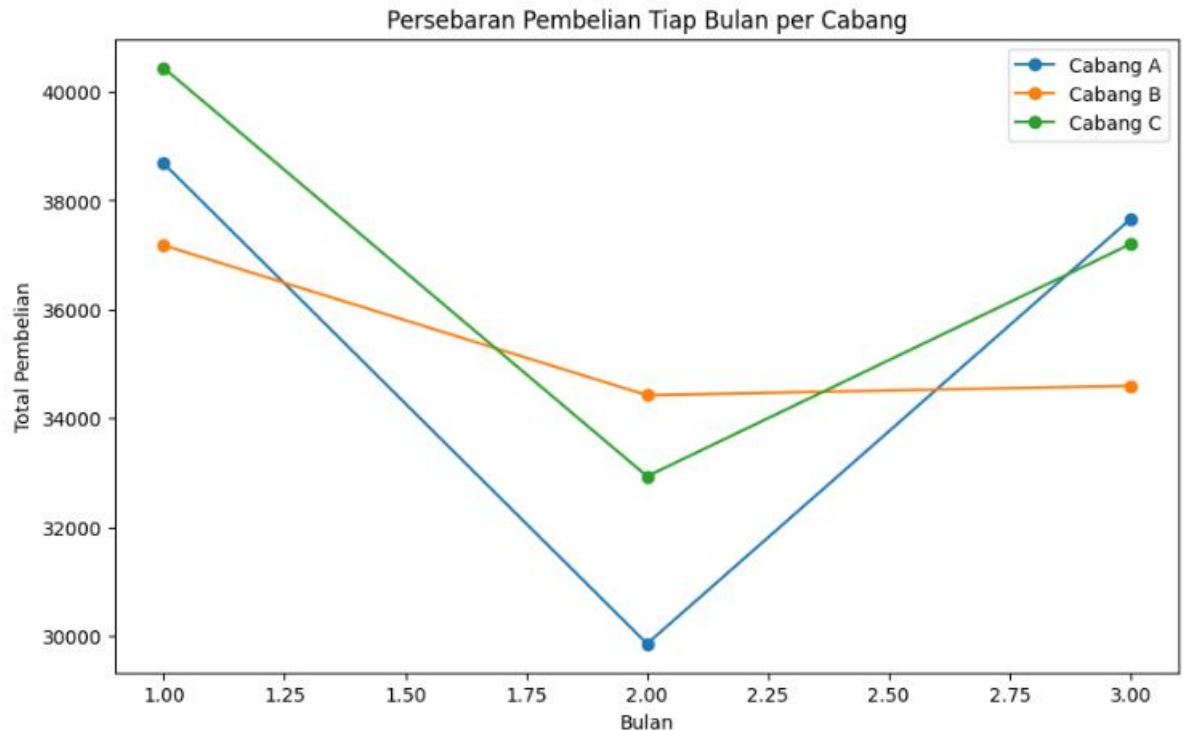
for branch in pembelian_per_bulan_percabang.columns:
    ax.plot(pembelian_per_bulan_percabang.index,
pembelian_per_bulan_percabang[branch], marker='o', label=f'Cabang {branch}')

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Bulan')
plt.ylabel('Total Pembelian')
plt.title('Persebaran Pembelian Tiap Bulan per Cabang')

# Menampilkan legenda
plt.legend()

# Menampilkan line chart
plt.show()

```



Penjelasan :

Kode di atas merupakan contoh implementasi untuk membuat line chart yang menunjukkan persebaran pembelian tiap bulan per cabang menggunakan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode `df['Date'] = pd.to_datetime(df['Date'])` digunakan untuk mengubah kolom 'Date' menjadi tipe data datetime agar dapat melakukan manipulasi tanggal. Kode `df['Month'] = df['Date'].dt.month` digunakan untuk mengekstraksi informasi bulan dari kolom 'Date' dan menyimpannya dalam kolom baru 'Month'.

Dengan menggunakan metode `'groupby'` pada kolom 'Month' dan 'Branch', kita mengelompokkan data berdasarkan bulan dan cabang. Kemudian, dengan menggunakan metode `'sum()'` pada kolom 'Total', kita menghitung total pembelian untuk setiap kombinasi bulan dan cabang. Hasil perhitungan ini disimpan dalam variabel `'pembelian_per_bulan_percabang'`.

Selanjutnya, kita menggunakan loop `'for'` untuk melakukan plot garis pada line chart untuk setiap cabang. Kode `ax.plot(pembelian_per_bulan_percabang.index, pembelian_per_bulan_percabang[branch], marker='o', label=f'Cabang {branch}')` digunakan untuk melakukan plot garis dengan menggunakan data dari `'pembelian_per_bulan_percabang'`. Parameter `'marker='o'` digunakan untuk menampilkan titik pada setiap data. Parameter `'label'` digunakan untuk memberikan label pada garis yang menunjukkan nama cabang.

## 12. Rata - rata rating pelanggan untuk setiap kategori produk

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

smoker = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung rata-rata rating pelanggan untuk setiap kategori produk
rating_rata = smoker.groupby('Product line')['Rating'].mean().reset_index()

# Menyiapkan data untuk bubble chart
kategori_produk = rating_rata['Product line']
rata_rating = rating_rata['Rating']
jumlah_penjualan = smoker.groupby('Product line')['Quantity'].sum()

# Membuat bubble chart
fig, ax = plt.subplots(figsize=(10, 6))
ax.scatter(kategori_produk, rata_rating, s=jumlah_penjualan/10, c='skyblue',
alpha=0.7)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Kategori Produk')
plt.ylabel('Rata-rata Rating')
plt.title('Rata-rata Rating Pelanggan untuk Setiap Kategori Produk')

# Menampilkan bubble chart
plt.show()
```

Penjelasan :

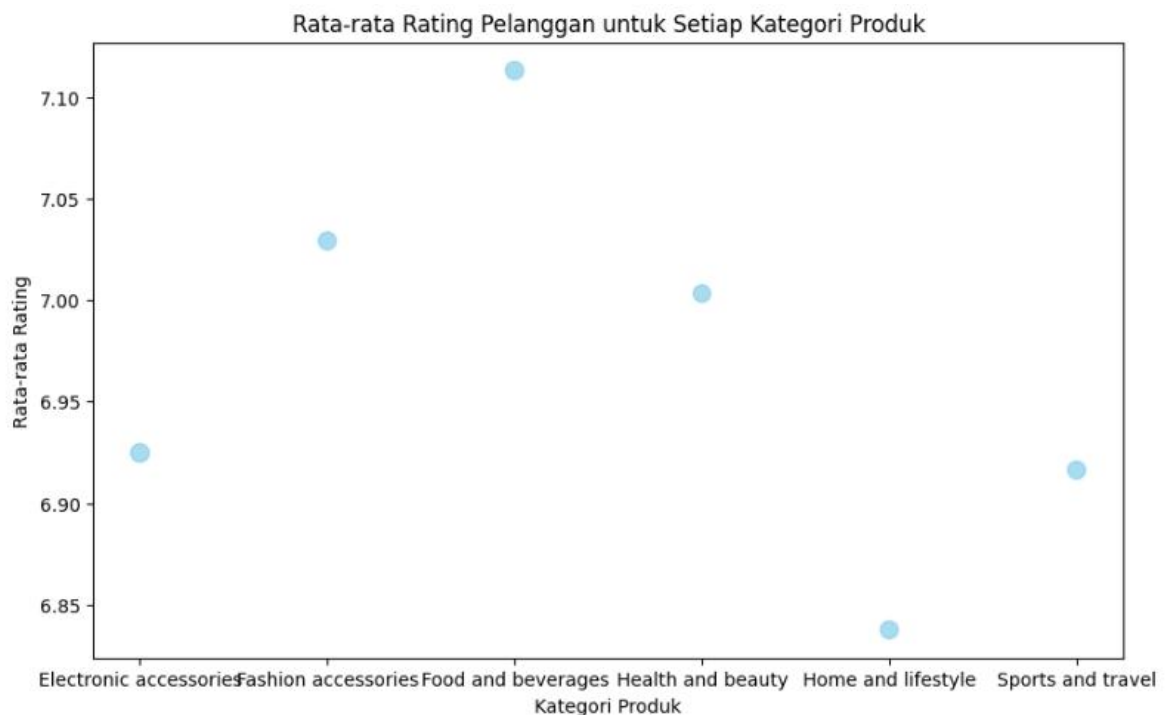
Kode di atas merupakan contoh implementasi untuk membuat bubble chart yang menunjukkan rata-rata rating pelanggan untuk setiap kategori produk menggunakan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya.

Dengan menggunakan metode `groupby` pada kolom 'Product line', kita mengelompokkan data berdasarkan kategori produk dan kemudian menghitung rata-rata rating pelanggan untuk setiap kategori produk. Hasil perhitungan ini disimpan dalam variabel `rating\_rata`.

Kode `kategori\_produk = rating\_rata['Product line']` digunakan untuk mengambil kolom 'Product line' dari DataFrame `rating\_rata` yang berisi kategori produk. Kode `rata\_rating = rating\_rata['Rating']` digunakan untuk mengambil kolom

'Rating' dari DataFrame `rating\_rate` yang berisi rata-rata rating pelanggan. Kode `jumlah\_penjualan = smoker.groupby('Product line')['Quantity'].sum()` digunakan untuk menghitung jumlah penjualan tiap kategori produk dengan menggunakan metode `groupby` pada kolom 'Product line' dan kemudian menjumlahkan kolom 'Quantity'. Hasil perhitungan ini disimpan dalam variabel `jumlah\_penjualan`.

Selanjutnya, kita menggunakan metode `scatter` pada objek sumbu `ax` untuk membuat bubble chart. Parameter pertama adalah data pada sumbu x, yaitu `kategori\_produk` dan `rata\_rating`. Parameter `s` digunakan untuk menentukan ukuran dari setiap bubble berdasarkan `jumlah\_penjualan`. Pada contoh ini, ukuran bubble diatur dengan membagi `jumlah\_penjualan` dengan 10. Parameter `c` digunakan untuk menentukan warna bubble, dalam hal ini digunakan warna 'skyblue'. Parameter `alpha` digunakan untuk mengatur tingkat transparansi dari bubble chart.



### 13. Jumlah Pelanggan berdasarkan jenis kelamin pada setiap kota

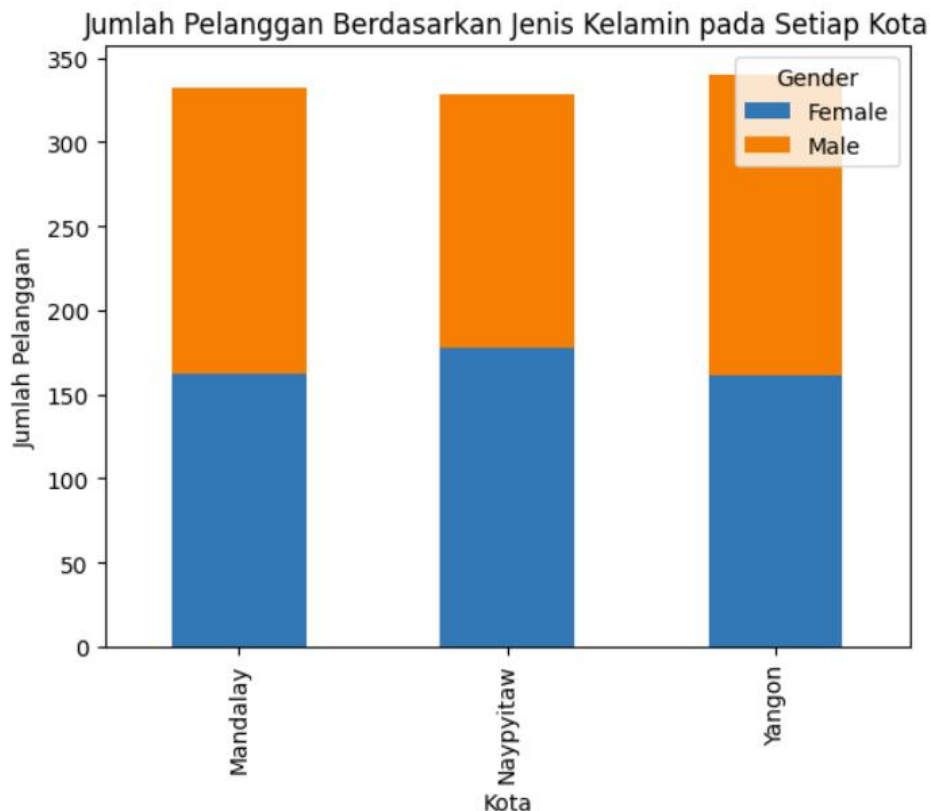
```
# menggabungkan data perkelompoknya dengan pivot
gender_city_table = pd.pivot_table(smoker, values='Quantity', index=['City'],
columns=['Gender'], aggfunc=len)

# membuat stacked bar chart
gender_city_table.plot(kind='bar', stacked=True)

# mengatur judul dan label sumbu
plt.title('Jumlah Pelanggan Berdasarkan Jenis Kelamin pada Setiap Kota')
plt.xlabel('Kota')
plt.ylabel('Jumlah Pelanggan')
```



```
# menampilkan plot  
plt.show()
```



Penjelasan:

Pada bagian ini akan ditampilkan jumlah pelanggan berdasarkan jenis kelamin di setiap kota, dimana pada dataset terdapat kolom gender, dan city yang akan digunakan untuk melakukan visualisasi tersebut. pada kode tersebut akan menggabungkan beberapa data dengan menggunakan pivot, yang terdiri dari 2 kolom yang diambil yaitu city dan gender, lalu akan ditampilkan dengan bar chart. Sehingga dihasilkan visualisasi data seperti gambar diatas yang berarti pada kota Mandalay terdapat 160 pelanggan berjenis kelamin wanita dan 330 pelanggan berjenis kelamin pria, pada kota Naypyitaw terdapat 170 pelanggan berjenis kelamin wanita dan 320 pelanggan berjenis kelamin pria, dan terakhir pada kota Yangon terdapat 160 pelanggan berjenis kelamin wanita, dan 340 pelanggan berjenis kelamin pria. grafik tersebut diberi 2 warna untuk membedakan masing-masing jenis kelamin, yaitu warna oren untuk jenis kelamin pria, dan biru jenis kelamin wanita.

14. Persentase penjualan setiap kategori produk pada setiap cabang/kota

```
import pandas as pd
```

```

import matplotlib.pyplot as plt

# membaca file csv
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# menghitung total penjualan setiap kategori produk pada setiap cabang/kota
penjualan_per_cabang = df.groupby(['Branch', 'Product line'])['Total'].sum().unstack()

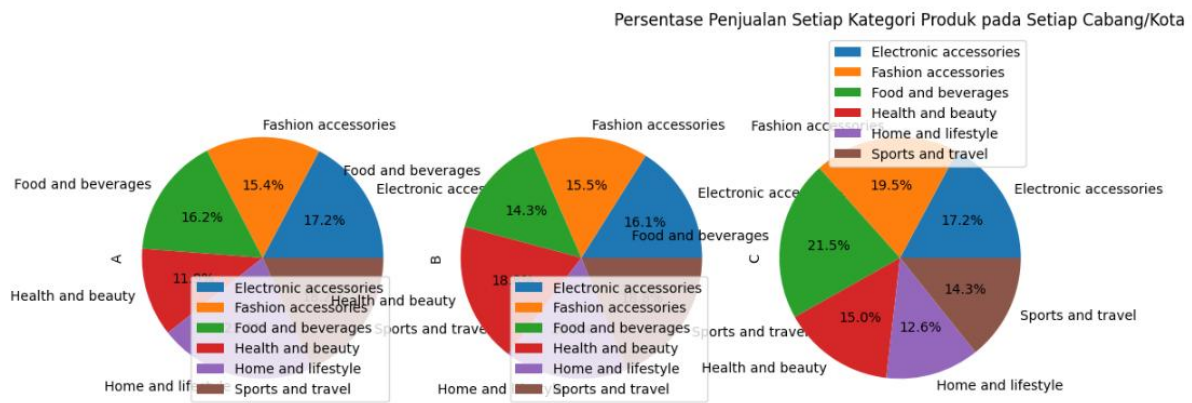
# menampilkan pie chart
penjualan_per_cabang.T.plot(kind='pie',          subplots=True,          figsize=(12, 6),
autopct='%1.1f%%')
plt.axis('equal')
plt.title('Persentase Penjualan Setiap Kategori Produk pada Setiap Cabang/Kota')
plt.legend(loc='upper right')
plt.show()

```

Penjelasan :

Pada bagian ini akan ditampilkan Persentase Penjualan Setiap Kategori Produk pada Setiap Cabang/Kota. Pada dataset terdapat kolom *branch* dan *product line* yang akan digunakan untuk menghasilkan visualisasi tersebut. kemudian menghitung total penjualan ('Total') untuk setiap kombinasi cabang/kota dan kategori produk. Hasilnya disusun dalam bentuk tabel dengan indeks 'Branch' dan kolom-kolom adalah kategori produk. Kemudian, membuat pie chart dari data penjualan\_per\_cabang. Baris kode penjualan\_per\_cabang.T.plot(kind='pie', subplots=True, figsize=(12, 6), autopct='%1.1f%%') membuat pie chart dengan menggunakan data penjualan\_per\_cabang. Argumen kind='pie' menentukan jenis chart yang dibuat adalah pie chart. subplots=True memastikan setiap kategori produk dihasilkan menjadi pie chart yang terpisah. figsize=(12, 6) mengatur ukuran figure untuk menampilkan pie chart. autopct='%1.1f%%' menampilkan persentase penjualan pada setiap sektor pie chart.

Selanjutnya, plt.axis('equal') digunakan untuk memastikan pie chart memiliki proporsi yang benar. plt.title('Persentase Penjualan Setiap Kategori Produk pada Setiap Cabang/Kota') memberikan judul pada pie chart. plt.legend(loc='upper right') menampilkan legenda pada pie chart, dengan posisi di sudut kanan atas. Terakhir, plt.show() digunakan untuk menampilkan pie chart yang telah dibuat. Sehingga hasil visualisasi menggunakan *pie chart* seperti berikut :



### 15. Total biaya pembelian pada tiap jenis produk untuk setiap jenis kelamin

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung total biaya pembelian pada tiap jenis produk untuk setiap jenis kelamin
biaya_per_produk_per_kelamin = df.groupby(['Product line', 'Gender'])['Total'].sum().unstack()

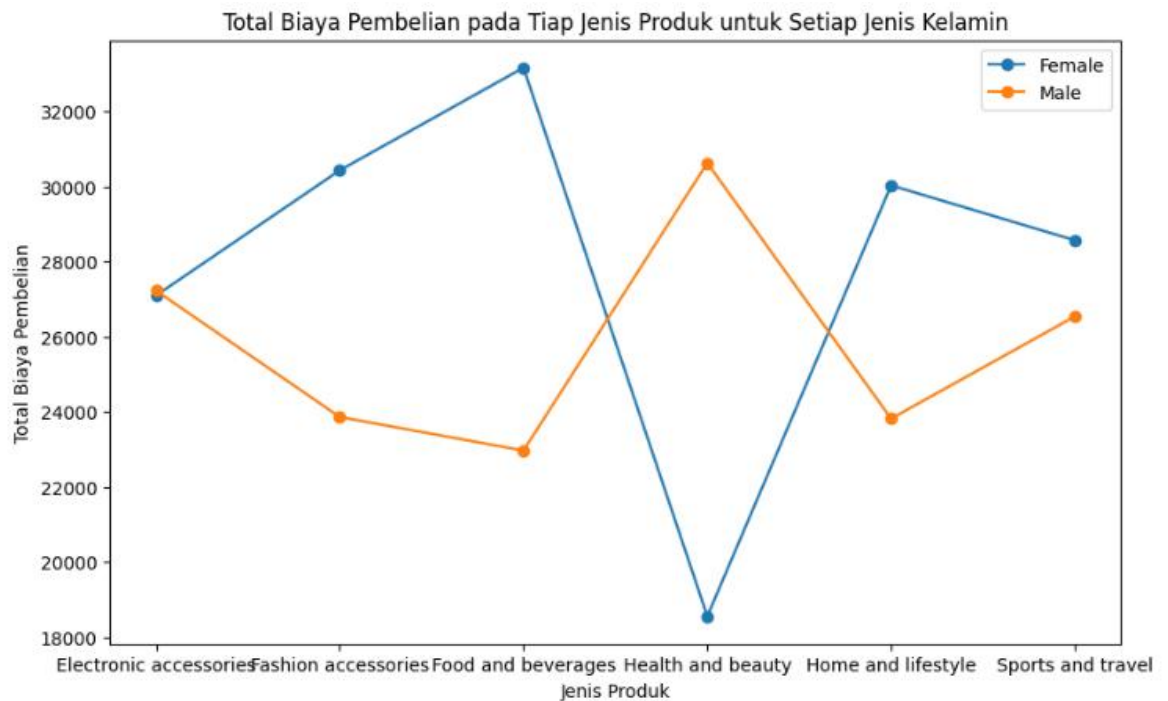
# Membuat line chart
fig, ax = plt.subplots(figsize=(10, 6))

# Menampilkan line chart untuk setiap jenis kelamin
for gender in biaya_per_produk_per_kelamin.columns:
    ax.plot(biaya_per_produk_per_kelamin.index,
            biaya_per_produk_per_kelamin[gender], marker='o', label=gender)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Jenis Produk')
plt.ylabel('Total Biaya Pembelian')
plt.title('Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Jenis Kelamin')

# Menampilkan legenda
plt.legend()

# Menampilkan line chart
plt.show()
```



Penjelasan:

Kode di atas merupakan contoh implementasi untuk membuat line chart yang menunjukkan total biaya pembelian pada tiap jenis produk untuk setiap jenis kelamin menggunakan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Dengan menggunakan metode `'groupby'` pada kolom `'Product line'` dan `'Gender'`, kita mengelompokkan data berdasarkan jenis produk dan jenis kelamin, dan kemudian menghitung total biaya pembelian menggunakan kolom `'Total'`. Hasil perhitungan ini disimpan dalam variabel `'biaya_per_produk_per_kelamin'`.

Selanjutnya, kita menggunakan metode `'plot'` pada objek sumbu `'ax'` untuk membuat line chart. Dalam loop `'for'`, kita menampilkan line chart untuk setiap jenis kelamin. Parameter pertama adalah data pada sumbu x, yaitu `'biaya_per_produk_per_kelamin.index'` yang merupakan jenis produk. Parameter kedua adalah data pada sumbu y, yaitu `'biaya_per_produk_per_kelamin[gender]'` yang merupakan total biaya pembelian. Parameter `'marker='o'` digunakan untuk menampilkan tanda bulat pada setiap titik data. Parameter `'label=gender'` digunakan untuk memberikan label pada line chart berdasarkan jenis kelamin.

Kode `'plt.xlabel('Jenis Produk')'` digunakan untuk menentukan label sumbu x dengan teks "Jenis Produk". Kode `'plt.ylabel('Total Biaya Pembelian')'` digunakan untuk menentukan label sumbu y dengan teks "Total Biaya Pembelian". Kode `'plt.title('Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Jenis Kelamin')'` digunakan untuk memberikan judul pada grafik.

16. Persentase produk yang dibeli oleh pelanggan member

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

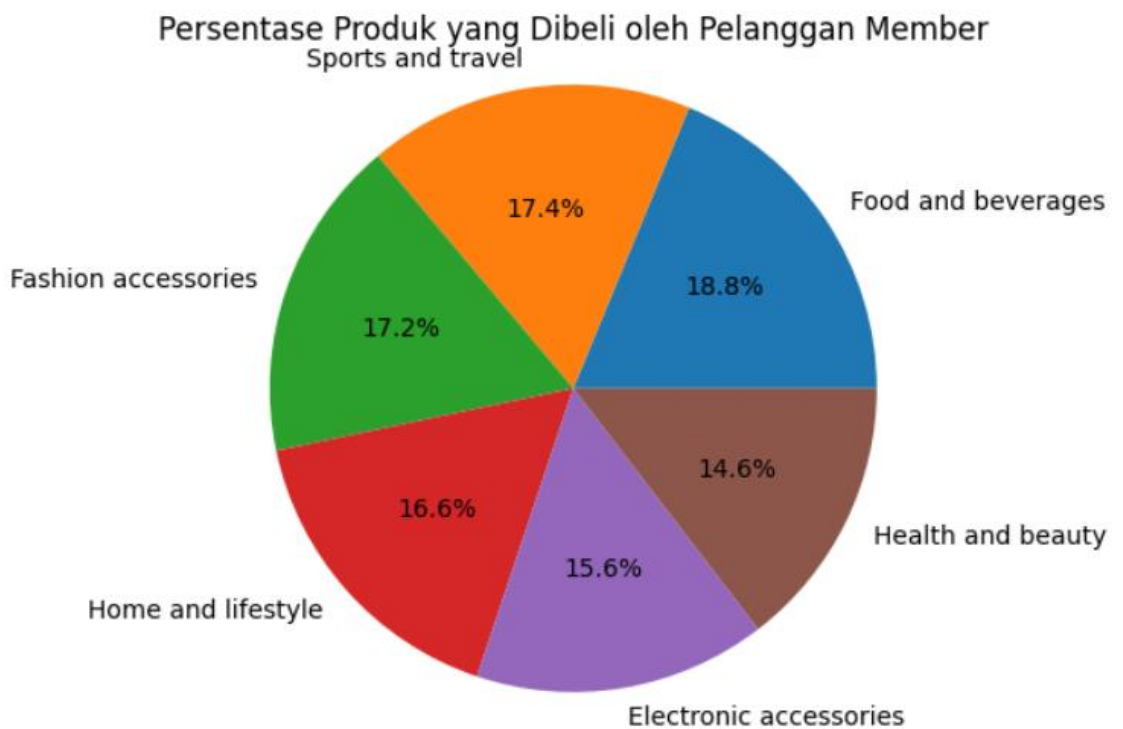
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung jumlah produk yang dibeli oleh pelanggan member
produk_member = df[df['Customer type'] == 'Member']['Product line'].value_counts()

# Menghitung persentase produk yang dibeli oleh pelanggan member
persentase_produk_member = (produk_member / produk_member.sum()) * 100

# Membuat pie chart persentase produk yang dibeli oleh pelanggan member
plt.pie(persentase_produk_member, labels=persentase_produk_member.index,
        autopct='%1.1f%%')
plt.title('Persentase Produk yang Dibeli oleh Pelanggan Member')
plt.axis('equal')

# Menampilkan pie chart
plt.show()
```



Penjelasan:

Kode di atas merupakan contoh implementasi untuk membuat pie chart yang menunjukkan persentase produk yang dibeli oleh pelanggan member menggunakan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Dengan menggunakan filter `df['Customer type'] == 'Member'`, kita memilih baris-baris dalam dataset yang memiliki nilai 'Customer type' adalah 'Member'. Kemudian, menggunakan metode `value_counts()`, kita menghitung jumlah produk yang dibeli oleh pelanggan member. Hasil perhitungan ini disimpan dalam variabel `produk_member`.

Kita menghitung persentase produk yang dibeli oleh pelanggan member dengan membagi jumlah produk yang dibeli oleh pelanggan member dengan jumlah total produk yang dibeli oleh semua pelanggan, kemudian dikalikan 100. Persentase ini disimpan dalam variabel `persentase_produk_member`.

#### 17. Persentase produk yang dibeli oleh pelanggan non-member

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

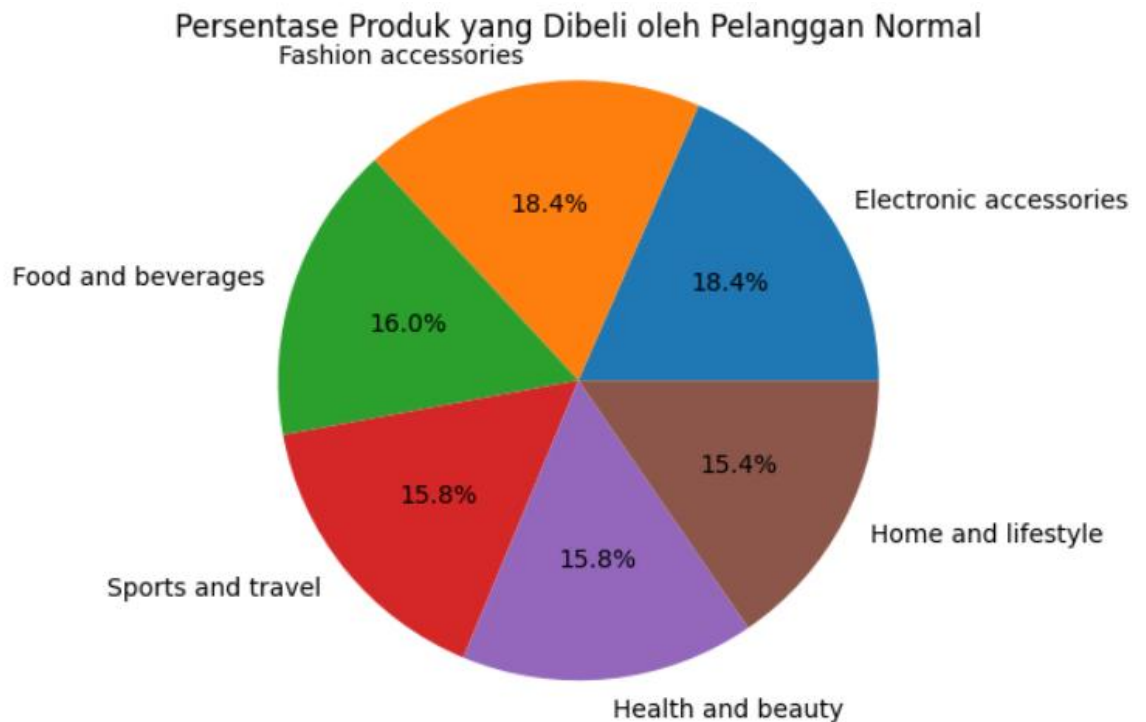
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung jumlah produk yang dibeli oleh pelanggan normal
produk_normal = df[df['Customer type'] == 'Normal']['Product line'].value_counts()

# Menghitung persentase produk yang dibeli oleh pelanggan normal
persentase_produk_normal = (produk_normal / produk_normal.sum()) * 100

# Membuat pie chart persentase produk yang dibeli oleh pelanggan normal
plt.pie(persentase_produk_normal, labels=persentase_produk_normal.index,
autopct='%1.1f%%')
plt.title('Persentase Produk yang Dibeli oleh Pelanggan Normal')
plt.axis('equal')

# Menampilkan pie chart
plt.show()
```



Penjelasan:

Kode di atas merupakan contoh implementasi untuk membuat pie chart yang menunjukkan persentase produk yang dibeli oleh pelanggan normal menggunakan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode `df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')` digunakan untuk membaca dataset yang berada dalam file CSV. Pastikan bahwa file CSV dengan nama "Kelompok 1\_supermarket sales\_csv.csv" tersedia di direktori yang sama dengan skrip Python.

Dengan menggunakan filter `df['Customer type'] == 'Normal'`, kita memilih baris-baris dalam dataset yang memiliki nilai 'Customer type' adalah 'Normal'. Kemudian, menggunakan metode `value_counts()`, kita menghitung jumlah produk yang dibeli oleh pelanggan normal. Hasil perhitungan ini disimpan dalam variabel `'produk_normal'`.

Kita menghitung persentase produk yang dibeli oleh pelanggan normal dengan membagi jumlah produk yang dibeli oleh pelanggan normal dengan jumlah total produk yang dibeli oleh semua pelanggan, kemudian dikalikan 100. Persentase ini disimpan dalam variabel `'persentase_produk_normal'`.

Kode `plt.pie(persentase_produk_normal, labels=persentase_produk_normal.index, autopct='%1.1f%%')` digunakan untuk membuat pie chart dari data persentase produk yang dibeli oleh pelanggan normal. Parameter pertama adalah data persentase yang akan ditampilkan pada pie chart. Parameter kedua adalah label untuk setiap

bagian pie chart, yaitu `persentase\_produk\_normal.index` yang merupakan jenis produk. Parameter `autopct='%1.1f%%'` digunakan untuk menampilkan persentase pada setiap bagian pie chart.

#### 18. Jumlah transaksi pada setiap jam untuk setiap cabang/kota

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

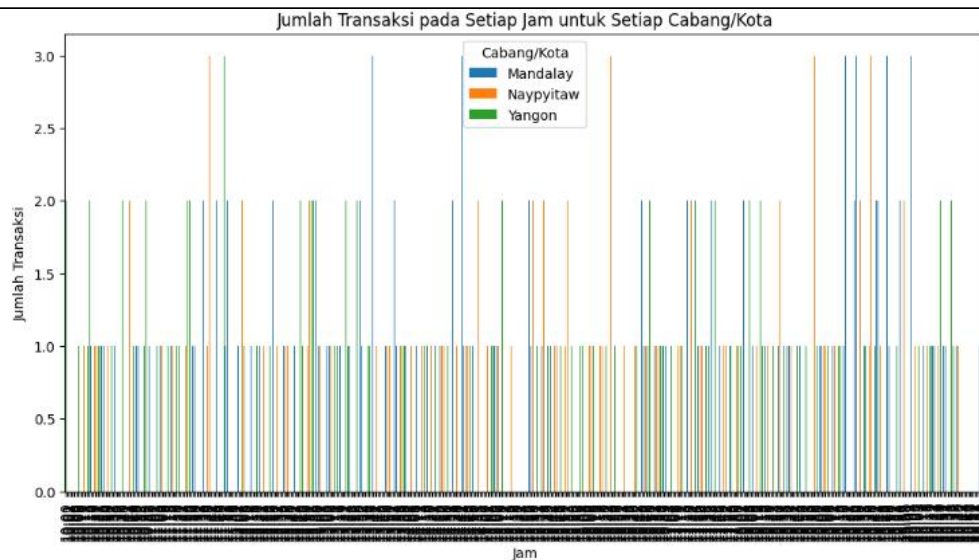
# Menghitung jumlah transaksi pada setiap jam untuk setiap cabang/kota
transaksi_per_jam = df.groupby(['Branch', 'City', 'Time'])['Invoice ID'].count().reset_index()

# Membuat bar chart
fig, ax = plt.subplots(figsize=(12, 6))
transaksi_per_jam.pivot_table(index='Time', columns='City', values='Invoice ID',
aggfunc='sum').plot(kind='bar', ax=ax)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Jam')
plt.ylabel('Jumlah Transaksi')
plt.title('Jumlah Transaksi pada Setiap Jam untuk Setiap Cabang/Kota')

# Menampilkan legend
plt.legend(title='Cabang/Kota')

# Menampilkan bar chart
plt.show()
```





Penjelasan :

Kode di atas menghitung jumlah transaksi pada setiap jam untuk setiap cabang/kota berdasarkan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode ``transaksi_per_jam = df.groupby(['Branch', 'City', 'Time'])['Invoice ID'].count().reset_index()`` digunakan untuk menghitung jumlah transaksi pada setiap jam untuk setiap cabang/kota. Kita menggunakan metode ``groupby()`` untuk mengelompokkan data berdasarkan kolom 'Branch', 'City', dan 'Time'. Kemudian, kita menghitung jumlah transaksi dengan menggunakan metode ``count()`` pada kolom 'Invoice ID'. Hasil perhitungan ini disimpan dalam DataFrame ``transaksi_per_jam``.

Kode ``fig, ax = plt.subplots(figsize=(12, 6))`` digunakan untuk membuat objek figure dan axes dengan ukuran tertentu. Kode ``transaksi_per_jam.pivot_table(index='Time', columns='City', values='Invoice ID', aggfunc='sum').plot(kind='bar', ax=ax)`` digunakan untuk membuat bar chart dari data jumlah transaksi pada setiap jam untuk setiap cabang/kota. Kita menggunakan metode ``pivot_table()`` untuk mengubah data menjadi format yang sesuai untuk bar chart. Parameter ``index`` adalah kolom 'Time' yang akan menjadi sumbu x.

Parameter ``columns`` adalah kolom 'City' yang akan menjadi grup untuk setiap batang pada sumbu x. Parameter ``values`` adalah kolom 'Invoice ID' yang akan digunakan untuk menghitung jumlah transaksi. Parameter ``aggfunc`` adalah 'sum' yang berarti kita akan menjumlahkan nilai pada kolom 'Invoice ID' untuk setiap grup. Kemudian, kita menggunakan metode ``plot()`` dengan ``kind='bar'`` untuk membuat bar chart. Parameter ``ax=ax`` digunakan untuk menentukan axes yang digunakan.

#### 19. Total biaya pembelian pada setiap kategori produk untuk setiap cabang/kota

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung total biaya pembelian pada setiap kategori produk untuk setiap cabang/kota
total_biaya = df.groupby(['Branch', 'City', 'Product line'])['cogs'].sum().unstack().reset_index()

# Membuat line chart
fig, ax = plt.subplots(figsize=(12, 6))

# Menampilkan line chart untuk setiap kategori produk dan cabang/kota
```

```

for i, row in total_biaya.iterrows():
    branch = row['Branch']
    city = row['City']
    plt.plot(row[2:], label=f'{branch} - {city}')

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Kategori Produk')
plt.ylabel('Total Biaya Pembelian')
plt.title('Total Biaya Pembelian pada Setiap Kategori Produk untuk Setiap Cabang/Kota')

# Menampilkan legenda
plt.legend()

# Menampilkan line chart
plt.show()

```

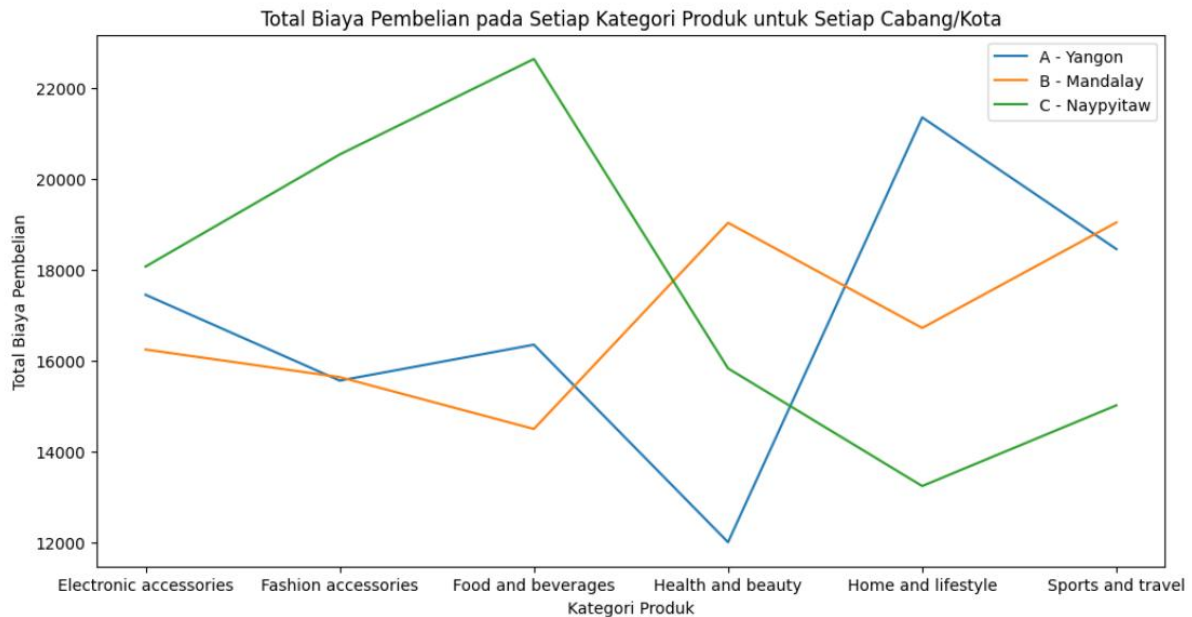
Penjelasan :

Pada bagian ini akan ditampilkan Total biaya pembelian pada setiap kategori produk untuk setiap cabang/kota. Pada dataset terdapat kolom *branch*, *city*, dan *product line* yang akan digunakan untuk menghasilkan visualisasi tersebut. menghitung total biaya pembelian pada setiap kategori produk untuk setiap cabang/kota. Baris kode `total_biaya = df.groupby(['Branch', 'City', 'Productline'])['cogs'].sum().unstack().reset_index()` mengelompokkan data berdasarkan kolom 'Branch', 'City', dan 'Product line', kemudian menghitung total biaya pembelian ('cogs') untuk setiap kombinasi cabang/kota dan kategori produk. Hasilnya disusun dalam bentuk tabel dengan indeks 'Branch' dan 'City', serta kolom-kolom adalah kategori produk.

Kemudian, membuat line chart dari data `total_biaya`. Baris kode `fig, ax = plt.subplots(figsize=(12, 6))` digunakan untuk membuat objek figure dan axis dengan ukuran 12x6 inci. Selanjutnya, dengan menggunakan loop `for i, row in total_biaya.iterrows():`, kita melakukan iterasi untuk setiap baris dalam `total_biaya`. Kita mendapatkan nilai cabang/kota dari kolom 'Branch' dan 'City' pada setiap baris. Kemudian, dengan `plt.plot(row[2:], label=f'{branch} - {city}')`, kita menggambar line chart untuk setiap baris, dimulai dari kolom ketiga (indeks ke-2) hingga akhir kolom. Label pada setiap line chart dibentuk dengan menggunakan kombinasi cabang/kota.

Setelah selesai menggambar line chart untuk setiap baris, kita menentukan label sumbu x dengan `plt.xlabel('Kategori Produk')`, label sumbu y dengan `plt.ylabel('Total Biaya Pembelian')`, dan judul chart dengan `plt.title('Total Biaya Pembelian pada Setiap Kategori Produk untuk Setiap Cabang/Kota')`. Kemudian,

menampilkan legenda menggunakan `plt.legend()` untuk menampilkan legenda dengan informasi cabang/kota. Terakhir, dengan `plt.show()`, kita menampilkan line chart yang telah dibuat sehingga menghasilkan visualisasi sebagai berikut :



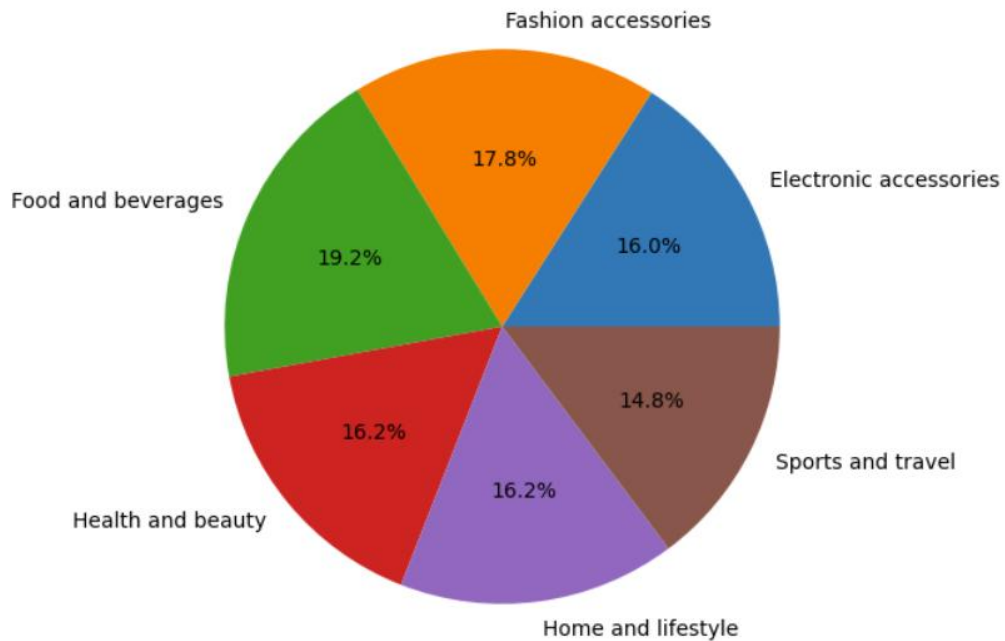
20. Persentase rating pelanggan yang memberikan rating 7 ke atas untuk setiap kategori produk

```
# mengambil data rating yang besar atau sama dengan 7
rating = smoker[smoker['Rating'] >= 7]

# hitung persentase rating untuk setiap kategori produk
rating_percentages = rating.groupby('Product line')['Rating'].count() /
len(rating) * 100

# plot pie chart
plt.figure(figsize=(8, 6))
plt.pie(rating_percentages, labels=rating_percentages.index,
autopct='%1.1f%%')
plt.title('Persentase Rating 7 ke atas untuk Setiap Kategori Produk')
plt.show()
```

Persentase Rating 7 ke atas untuk Setiap Kategori Produk



Penjelasan:

Pada bagian ini akan ditampilkan persentase rating 7 ke atas untuk masing-masing kategori produk. Pada dataset terdapat kolom rating dan product line yang akan digunakan untuk menghasilkan visualisasi tersebut. Pada kode diatas pertama harus mengambil seluruh data yang memiliki rating 7 atau lebih besar dari 7, dan data tersebut disimpan pada sebuah variabel bernama rating. Lalu akan dilakukan perhitungan persentase dan akan ditampilkan dalam bentuk pie chart. Sehingga dihasilkan visualisasi data seperti gambar diatas, yang berarti didapatkan data yaitu sebagai berikut:

- Fashion accessories memiliki 17,8 % produk yang rating 7 ke atas
- Electronic accessories memiliki 16.0 % produk yang rating 7 ke atas
- Sports and travel memiliki 14.8 % produk yang rating 7 ke atas
- Home and lifestyle memiliki 16.2 % produk yang rating 7 ke atas
- Health and beauty memiliki 16.2 % produk yang rating 7 ke atas
- Food and beverages memiliki 19.2 % produk yang rating 7 ke atas

21. Jumlah transaksi untuk setiap kategori produk untuk setiap cabang/kota

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

smoker = pd.read_csv('Kelompok 1_supermarket sales.csv')
```

```
# menghitung jumlah produk yang dibeli berdasarkan jenis kelamin dan kategori
produk
produk_per_gender = smoker.groupby(['City', 'Product line']).size().unstack()

# menampilkan column chart
produk_per_gender.plot(kind='bar')
plt.xlabel('Kota')
plt.ylabel('Jumlah Produk')
plt.title('Jumlah transaksi untuk setiap kategori produk untuk setiap cabang/kota')
plt.legend(title='Jenis Produk')
plt.xticks(rotation=45)
plt.show()
```

Penjelasan :

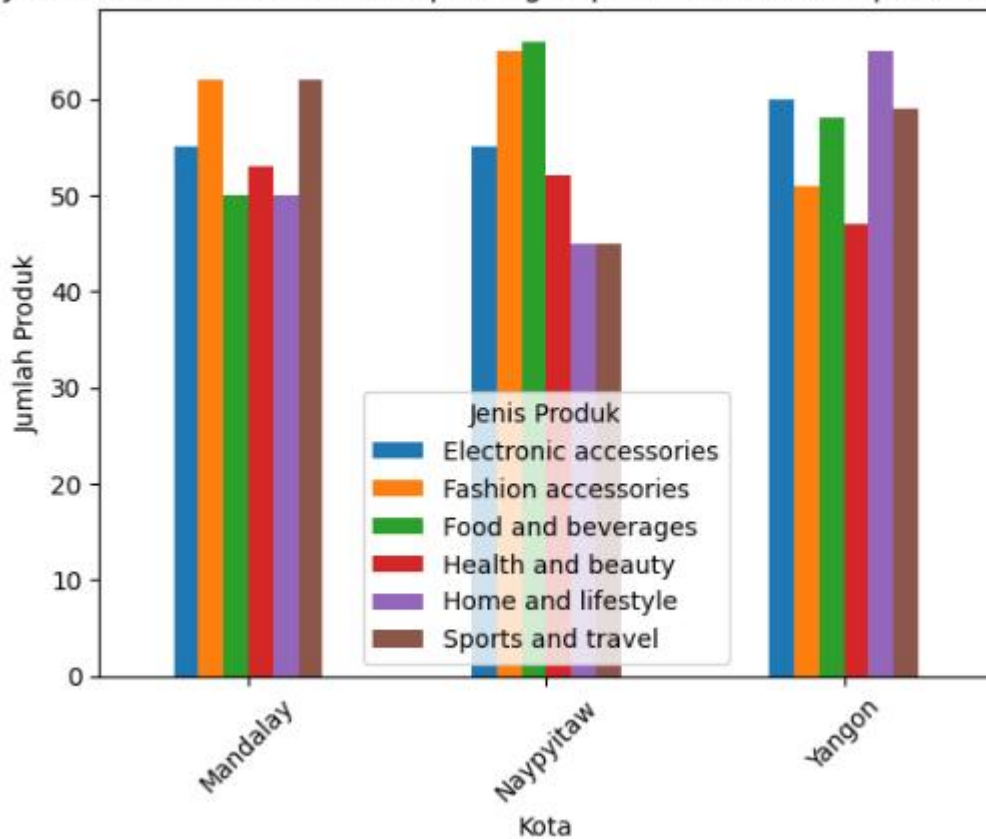
Kode di atas menghitung jumlah transaksi pada setiap jam untuk setiap cabang/kota berdasarkan dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode `transaksi_per_jam = df.groupby(['Branch', 'City', 'Time'])['Invoice ID'].count().reset_index()` digunakan untuk menghitung jumlah transaksi pada setiap jam untuk setiap cabang/kota. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Branch', 'City', dan 'Time'. Kemudian, kita menghitung jumlah transaksi dengan menggunakan metode `count()` pada kolom 'Invoice ID'. Hasil perhitungan ini disimpan dalam DataFrame `transaksi_per_jam`.

Kode `fig, ax = plt.subplots(figsize=(12, 6))` digunakan untuk membuat objek figure dan axes dengan ukuran tertentu. Kode `transaksi_per_jam.pivot_table(index='Time', columns='City', values='Invoice ID', aggfunc='sum').plot(kind='bar', ax=ax)` digunakan untuk membuat bar chart dari data jumlah transaksi pada setiap jam untuk setiap cabang/kota. Kita menggunakan metode `pivot_table()` untuk mengubah data menjadi format yang sesuai untuk bar chart. Parameter `index` adalah kolom 'Time' yang akan menjadi sumbu x.

Parameter `columns` adalah kolom 'City' yang akan menjadi grup untuk setiap batang pada sumbu x. Parameter `values` adalah kolom 'Invoice ID' yang akan digunakan untuk menghitung jumlah transaksi. Parameter `aggfunc` adalah 'sum' yang berarti kita akan menjumlahkan nilai pada kolom 'Invoice ID' untuk setiap grup. Kemudian, kita menggunakan metode `plot()` dengan `kind='bar'` untuk membuat bar chart. Parameter `ax=ax` digunakan untuk menentukan axes yang digunakan.

Jumlah transaksi untuk setiap kategori produk untuk setiap cabang/kota



22. Total biaya pembelian pada setiap jenis produk untuk setiap jenis

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung total biaya pembelian pada setiap jenis produk untuk setiap jenis
pembayaran
biaya_per_produk_per_pembayaran = df.groupby(['Product line',
'Payment'])['Total'].sum().unstack()

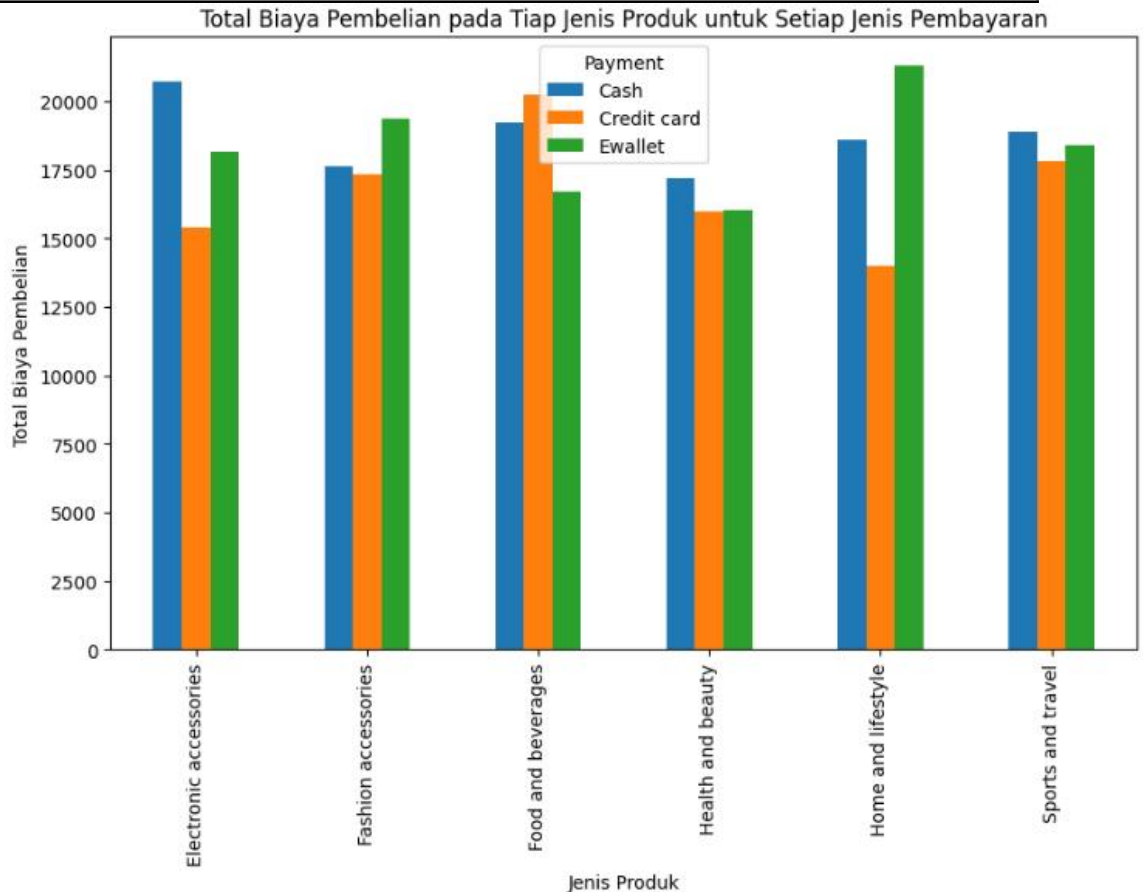
# Membuat bar chart
fig, ax = plt.subplots(figsize=(10, 6))

# Menampilkan bar chart untuk setiap jenis pembayaran
biaya_per_produk_per_pembayaran.plot(kind='bar', ax=ax)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Jenis Produk')
plt.ylabel('Total Biaya Pembelian')
plt.title('Total Biaya Pembelian pada Tiap Jenis Produk untuk Setiap Jenis
```

Pembayaran')

# Menampilkan bar chart  
plt.show()



Penjelasan:

Kode di atas menghitung total biaya pembelian pada setiap jenis produk untuk setiap jenis pembayaran pada dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode ``biaya_per_produk_per_pembayaran = df.groupby(['Product line', 'Payment'])['Total'].sum().unstack()`` digunakan untuk menghitung total biaya pembelian pada setiap jenis produk untuk setiap jenis pembayaran. Kita menggunakan metode ``groupby()`` untuk mengelompokkan data berdasarkan kolom 'Product line' dan 'Payment'. Kemudian, kita menggunakan metode ``sum()`` untuk menjumlahkan nilai kolom 'Total' pada setiap kelompok. Hasil perhitungan ini kita ubah menjadi bentuk tabel menggunakan metode ``unstack()`` sehingga kita memiliki kolom-kolom yang mewakili jenis pembayaran.

Kode `fig, ax = plt.subplots(figsize=(10, 6))` digunakan untuk membuat figure dan axes dengan ukuran 10x6. Kode `biaya_per_produk_per_pembayaran.plot(kind='bar', ax=ax)` digunakan untuk membuat bar chart dari data total biaya pembelian pada setiap jenis produk untuk setiap jenis pembayaran. Kita menggunakan metode `plot()` dengan `kind='bar'` untuk membuat bar chart. Setiap batang pada sumbu x akan mewakili jenis produk, dan tinggi batang akan menunjukkan total biaya pembelian. Setiap kolom pada chart akan mewakili jenis pembayaran yang berbeda.

### 23. total gross income per jenis kelamin

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

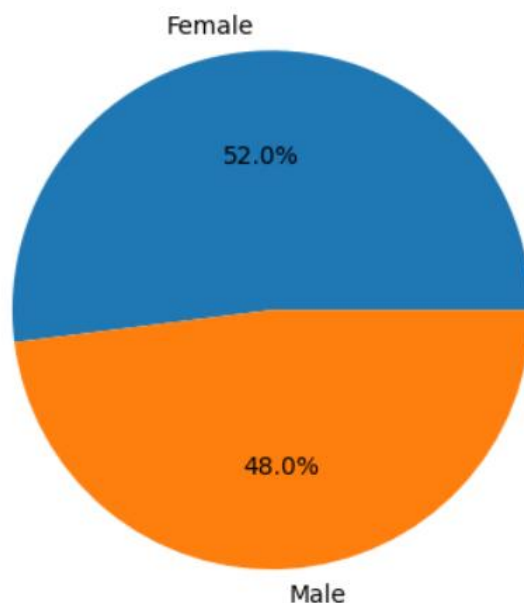
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total gross income per jenis kelamin
gross_income_per_gender = df.groupby('Gender')['gross income'].sum()

# Membuat pie chart
plt.pie(gross_income_per_gender, labels=gross_income_per_gender.index,
autopct='%1.1f%%')
plt.title('Total Gross Income per Jenis Kelamin')

# Menampilkan pie chart
plt.show()
```

Total Gross Income per Jenis Kelamin





Penjelasan:

Kode di atas menghitung total gross income (pendapatan kotor) per jenis kelamin pada dataset supermarket sales dan membuat pie chart untuk menampilkan informasi tersebut. Berikut adalah penjelasan langkah-langkahnya:

Kode `gross_income_per_gender = df.groupby('Gender')['gross income'].sum()` digunakan untuk menghitung total gross income per jenis kelamin. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Gender'. Kemudian, kita menggunakan metode `sum()` untuk menjumlahkan nilai kolom 'gross income' pada setiap kelompok.

Kode `plt.pie(gross_income_per_gender, labels=gross_income_per_gender.index, autopct='%1.1f%%')` digunakan untuk membuat pie chart dari data total gross income per jenis kelamin. Kita menggunakan fungsi `pie()` dari modul `matplotlib.pyplot`. Argumen pertama adalah data yang akan ditampilkan dalam bentuk pie chart, yaitu `gross_income_per_gender`. Argumen kedua adalah label untuk setiap bagian pie chart, yaitu `gross_income_per_gender.index` yang berisi jenis kelamin. Argumen `autopct='%1.1f%%'` digunakan untuk menampilkan persentase pada setiap bagian pie chart.

#### 24. persebaran payment per jenis kelamin

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

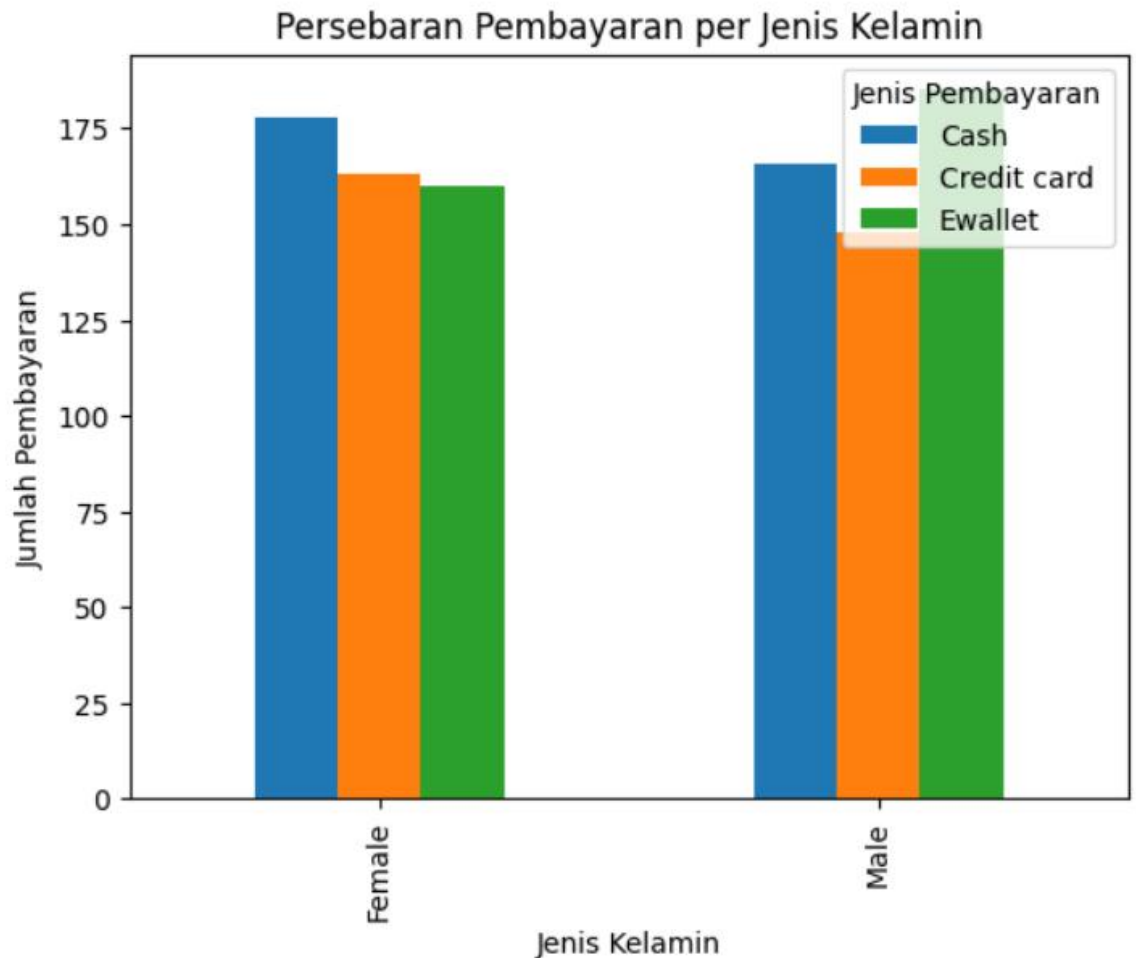
# Menghitung jumlah pembayaran per jenis kelamin
payment_per_gender = df.groupby(['Gender',
'Payment'])['Quantity'].count().unstack()

# Membuat bar chart
payment_per_gender.plot(kind='bar')

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Jenis Kelamin')
plt.ylabel('Jumlah Pembayaran')
plt.title('Persebaran Pembayaran per Jenis Kelamin')

# Menampilkan legend
plt.legend(title='Jenis Pembayaran')

# Menampilkan bar chart
plt.show()
```



Penjelasan:

Kode di atas menghitung jumlah pembayaran per jenis kelamin pada dataset supermarket sales dan membuat bar chart untuk menampilkan informasi tersebut. Berikut adalah penjelasan langkah-langkahnya:

Kode ``payment_per_gender = df.groupby(['Gender', 'Payment'])['Quantity'].count().unstack()`` digunakan untuk menghitung jumlah pembayaran per jenis kelamin. Kita menggunakan metode ``groupby()`` untuk mengelompokkan data berdasarkan kolom 'Gender' dan 'Payment'. Kemudian, kita menggunakan metode ``count()`` untuk menghitung jumlah entri dalam setiap kelompok. Akhirnya, metode ``unstack()`` digunakan untuk mengubah hasil perhitungan menjadi format yang lebih mudah digunakan dalam pembuatan bar chart.

## 25. total product line per jenis kelamin

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

# Membaca dataset
```

```

df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total product line untuk setiap jenis kelamin
total_product = df.groupby('Gender')['Product line'].nunique()

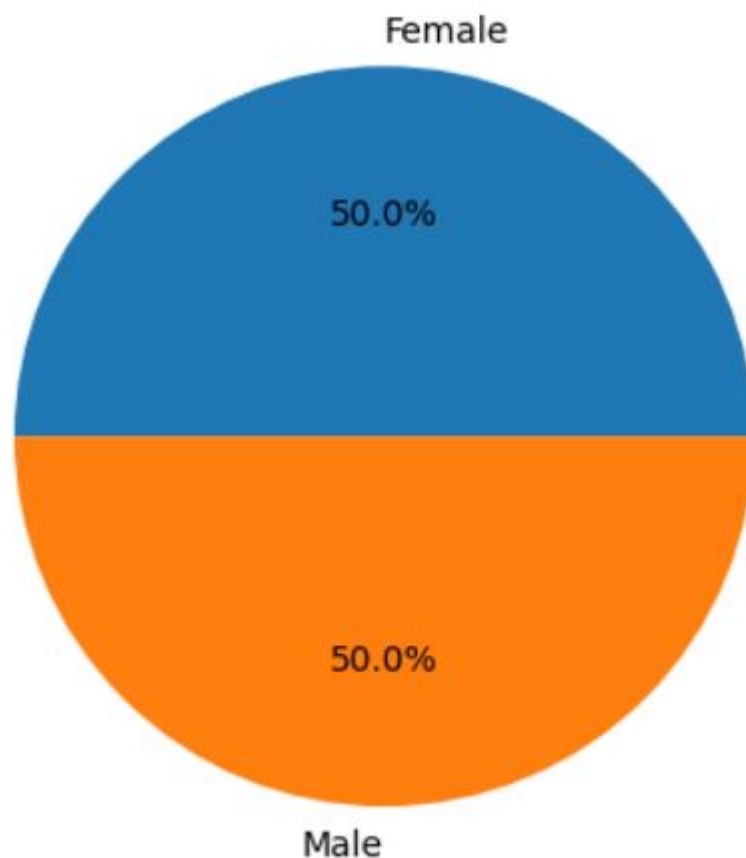
# Membuat pie chart
fig, ax = plt.subplots()
ax.pie(total_product, labels=total_product.index, autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Total Product Line per Jenis Kelamin')

# Menampilkan pie chart
plt.show()

```

### Total Product Line per Jenis Kelamin



Penjelasan:

Kode di atas menghitung jumlah pembayaran per jenis kelamin pada dataset supermarket sales dan membuat bar chart untuk menampilkan informasi tersebut. Berikut adalah penjelasan langkah-langkahnya:

Kode `payment\_per\_gender=df.groupby(['Gender','Payment'])['Quantity'].count().unstack()` digunakan untuk menghitung jumlah pembayaran per jenis kelamin. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Gender' dan 'Payment'. Kemudian, kita menggunakan metode `count()` untuk menghitung jumlah entri dalam setiap kelompok. Akhirnya, metode `unstack()` digunakan untuk mengubah hasil perhitungan menjadi format yang lebih mudah digunakan dalam pembuatan bar chart.

## 26. total pembelian per jenis kelamin

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

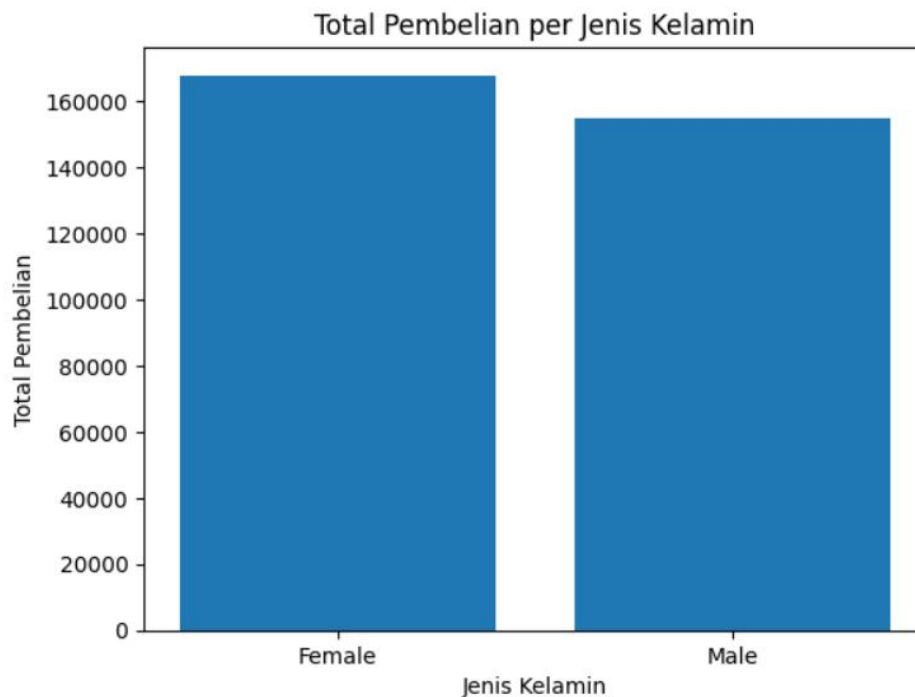
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total pembelian per jenis kelamin
total_pembelian = df.groupby('Gender')['Total'].sum()

# Membuat bar chart
fig, ax = plt.subplots()
ax.bar(total_pembelian.index, total_pembelian)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Jenis Kelamin')
plt.ylabel('Total Pembelian')
plt.title('Total Pembelian per Jenis Kelamin')

# Menampilkan bar chart
plt.show()
```



Penjelasan:

Kode `total_pembelian = df.groupby('Gender')['Total'].sum()` digunakan untuk menghitung total pembelian (kolom 'Total') untuk setiap jenis kelamin (kolom 'Gender'). Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Gender'. Kemudian, kita memilih kolom 'Total' dan menggunakan metode `sum()` untuk menjumlahkan nilai dalam setiap kelompok.

## 27. total quantity per jenis kelamin

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

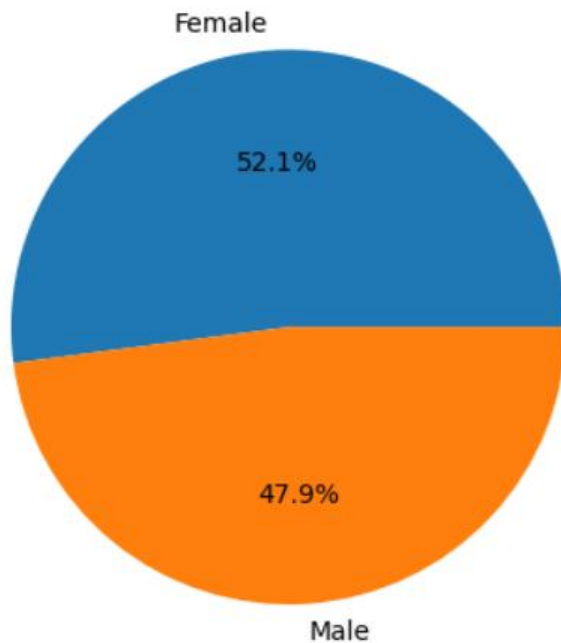
# Menghitung total quantity per jenis kelamin
total_quantity = df.groupby('Gender')['Quantity'].sum()

# Membuat pie chart
fig, ax = plt.subplots()
ax.pie(total_quantity, labels=total_quantity.index, autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Total Quantity per Jenis Kelamin')
```

```
# Menampilkan pie chart  
plt.show()
```

Total Quantity per Jenis Kelamin



Penjelasan:

Kode `total_quantity = df.groupby('Gender')['Quantity'].sum()` digunakan untuk menghitung total quantity (kolom 'Quantity') untuk setiap jenis kelamin (kolom 'Gender'). Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Gender'. Kemudian, kita memilih kolom 'Quantity' dan menggunakan metode `sum()` untuk menjumlahkan nilai dalam setiap kelompok.

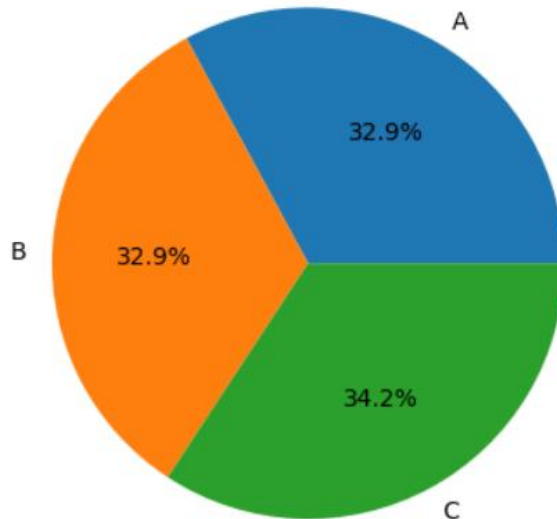
## 28. total tax percabang

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
# Membaca dataset  
df = pd.read_csv('Kelompok 1_supermarket sales.csv')  
  
# Menghitung total tax 5% per cabang  
df['tax_5_percent'] = df['Total'] * 0.05  
total_tax_5_percent = df.groupby('Branch')['tax_5_percent'].sum()  
  
# Membuat pie chart  
fig, ax = plt.subplots()  
ax.pie(total_tax_5_percent, labels=total_tax_5_percent.index, autopct='%1.1f%%')
```

```
# Menentukan judul chart
plt.title('Persentase Total Tax 5% per Cabang')

# Menampilkan pie chart
plt.show()
```

Persentase Total Tax 5% per Cabang



penjelasan:

Kode `df['tax_5_percent'] = df['Total'] * 0.05` digunakan untuk membuat kolom baru 'tax\_5\_percent' yang berisi nilai pajak 5% dari kolom 'Total' pada setiap transaksi. Nilai pajak dihitung dengan mengalikan nilai total transaksi dengan 0.05 (5%).

Kode `total_tax_5_percent = df.groupby('Branch')['tax_5_percent'].sum()` digunakan untuk menghitung total pajak 5% (kolom 'tax\_5\_percent') yang dikumpulkan oleh setiap cabang. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Branch'. Kemudian, kita memilih kolom 'tax\_5\_percent' dan menggunakan metode `sum()` untuk menjumlahkan nilai dalam setiap kelompok.

## 29. total tax per city

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total tax 5% per kota
```

```

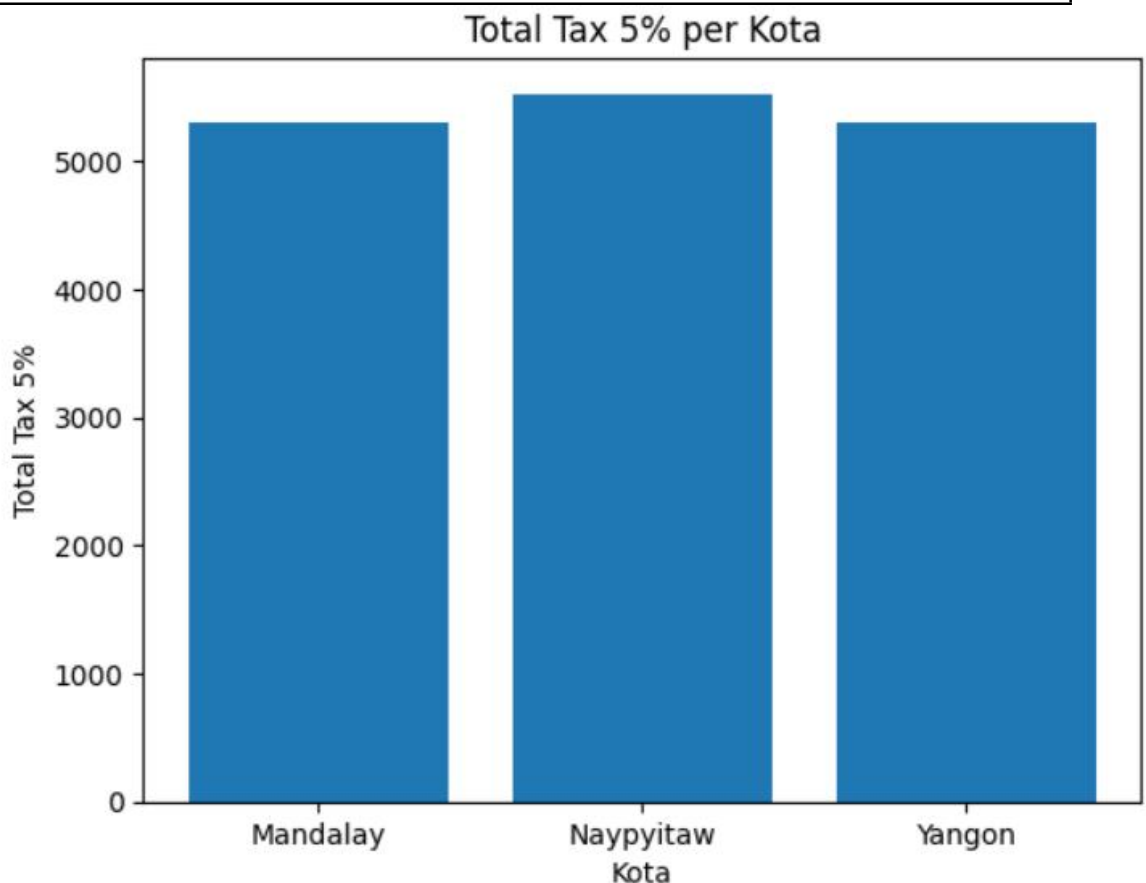
df['Tax_5percent'] = df['Total'] * 0.05
total_tax_5percent_per_city = df.groupby('City')['Tax_5percent'].sum()

# Membuat bar chart
fig, ax = plt.subplots()
ax.bar(total_tax_5percent_per_city.index, total_tax_5percent_per_city)

# Menentukan label sumbu x dan y, serta judul chart
plt.xlabel('Kota')
plt.ylabel('Total Tax 5%')
plt.title('Total Tax 5% per Kota')

# Menampilkan bar chart
plt.show()

```



Penjelasan:

Kode `df['Tax_5percent'] = df['Total'] * 0.05` digunakan untuk membuat kolom baru 'Tax\_5percent' yang berisi nilai pajak 5% dari kolom 'Total' pada setiap transaksi. Nilai pajak dihitung dengan mengalikan nilai total transaksi dengan 0.05 (5%).

Kode `total_tax_5percent_per_city = df.groupby('City')['Tax_5percent'].sum()` digunakan untuk menghitung total pajak 5%



(kolom 'Tax\_5percent') yang dikumpulkan oleh setiap kota. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'City'. Kemudian, kita memilih kolom 'Tax\_5percent' dan menggunakan metode `sum()` untuk menjumlahkan nilai dalam setiap kelompok.

### 30. total tax per gender

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

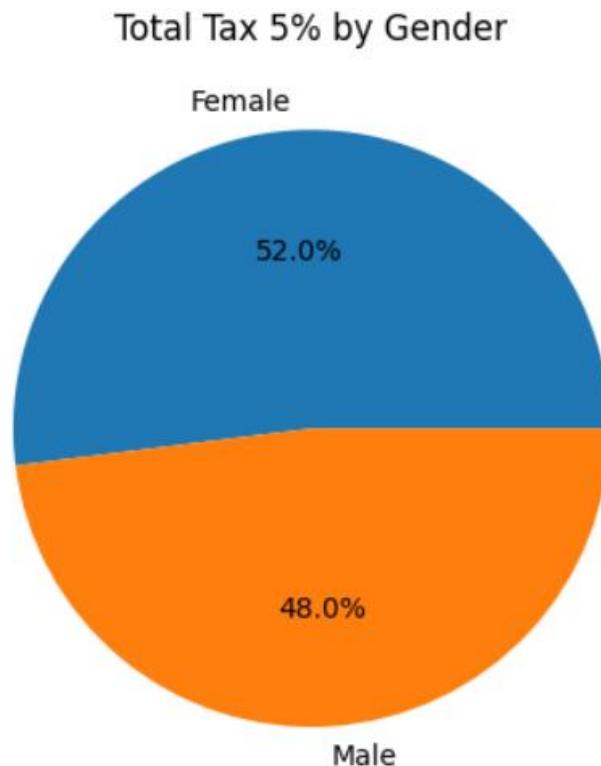
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total tax 5% per gender
df['Tax_5percent'] = df['Total'] * 0.05
total_tax_5percent_per_gender = df.groupby('Gender')['Tax_5percent'].sum()

# Membuat pie chart
fig, ax = plt.subplots()
ax.pie(total_tax_5percent_per_gender, labels=total_tax_5percent_per_gender.index,
autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Total Tax 5% by Gender')

# Menampilkan pie chart
plt.show()
```



Penjelasan:

`df['Tax_5percent'] = df['Total'] * 0.05`: Pada baris pertama, kita membuat kolom baru bernama 'Tax\_5percent' di DataFrame 'df'. Kolom ini akan berisi hasil dari perkalian kolom 'Total' dengan 0.05, yang mewakili pajak 5% dari total.

`df.groupby('Gender')['Tax_5percent'].sum()`: Perintah ini mengelompokkan baris-baris dalam DataFrame 'df' berdasarkan nilai pada kolom 'Gender'. Kemudian, dari setiap kelompok, kita memilih kolom 'Tax\_5percent'. Metode `sum()` diterapkan pada kolom 'Tax\_5percent' yang dipilih untuk menghitung total dari semua nilai pajak 5% dalam setiap kelompok gender.

### 31. total tax per product line

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Menghitung total tax 5% per product line
df['Tax_5percent'] = df['Total'] * 0.05
total_tax_5percent_per_product_line = df.groupby('Product
line')['Tax_5percent'].sum()

# Mengambil data product line dan total tax 5% per product line
```

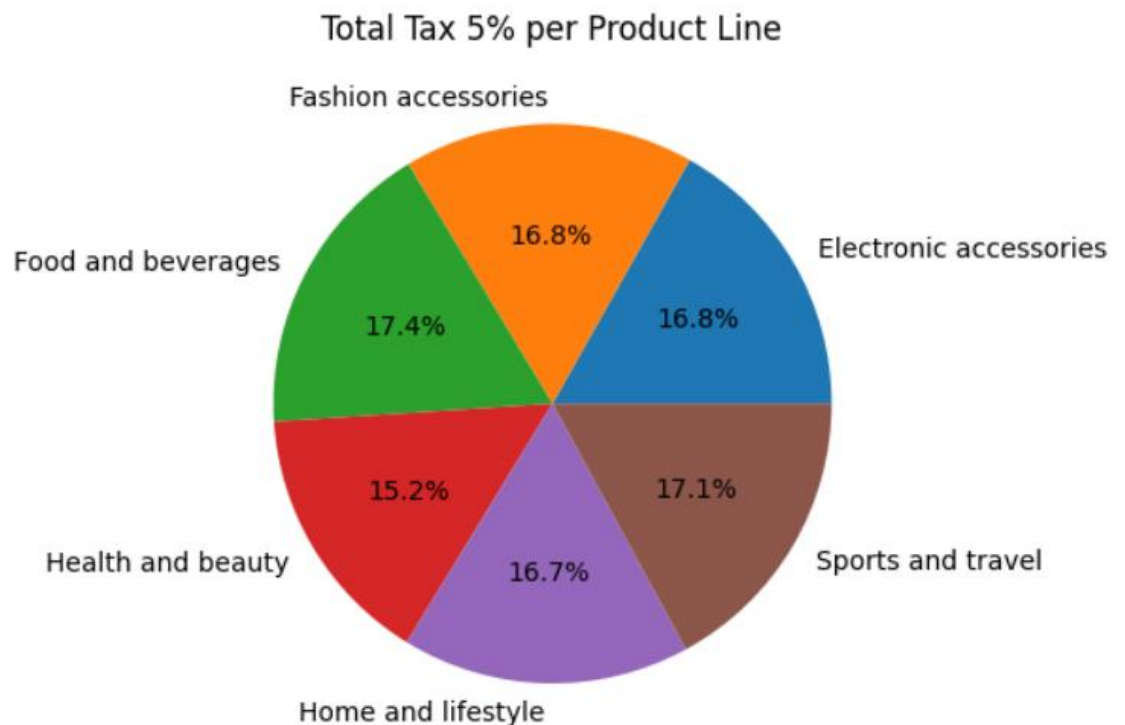
```

product_line = total_tax_5percent_per_product_line.index
total_tax_5percent = total_tax_5percent_per_product_line.values

# Membuat pie chart
plt.pie(total_tax_5percent, labels=product_line, autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Total Tax 5% per Product Line')

```



Penjelasan:

Kode di atas digunakan untuk menghitung total pajak 5% per jenis kelamin pada dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode `df['Tax_5percent'] = df['Total'] * 0.05` digunakan untuk membuat kolom baru 'Tax\_5percent' yang berisi nilai pajak 5% dari kolom 'Total' pada setiap transaksi. Nilai pajak dihitung dengan mengalikan nilai total transaksi dengan 0.05 (5%).

Kode `total_tax_5percent_per_gender=df.groupby('Gender')['Tax_5percent'].sum()` digunakan untuk menghitung total pajak 5% (kolom 'Tax\_5percent') yang

dikumpulkan oleh setiap jenis kelamin. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan kolom 'Gender'. Kemudian, kita memilih kolom 'Tax\_5percent' dan menggunakan metode `sum()` untuk menjumlahkan nilai dalam setiap kelompok.

### 32. Total Gross Income Terbanyak Tiap Branch

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

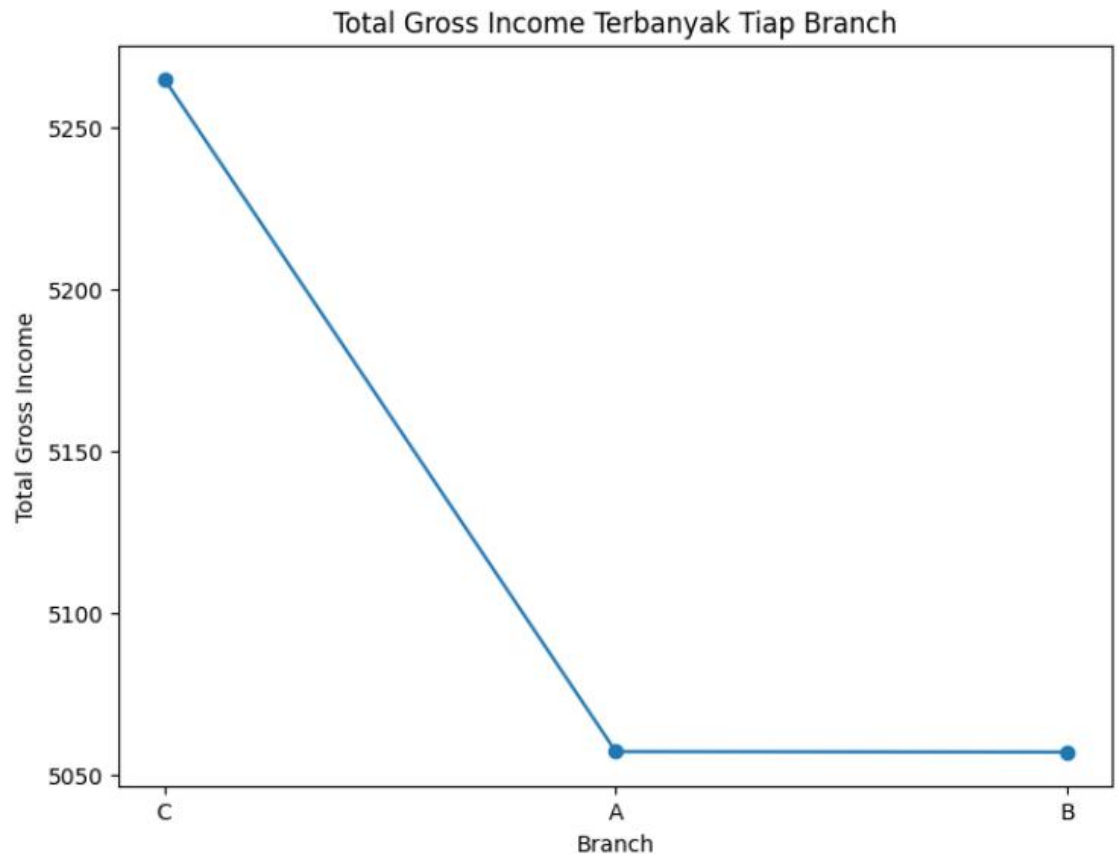
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# kelompokkan data berdasarkan branch dan jumlah total gross income
branch_income = df.groupby('Branch')['gross income'].sum().sort_values(ascending=False)

# hasil dalam bentuk tulisan
print(branch_income)

# create line chart
plt.figure(figsize=(8, 6))
plt.plot(branch_income.index, branch_income.values, marker='o')
plt.title('Total Gross Income Terbanyak Tiap Branch')
plt.xlabel('Branch')
plt.ylabel('Total Gross Income')
plt.show()
```

```
Branch
C    5265.1765
A    5057.1605
B    5057.0320
Name: gross income, dtype: float64
```



Penjelasan:

Kode di atas mengelompokkan data berdasarkan cabang (branch) dan menghitung jumlah total gross income (pendapatan kotor) pada dataset supermarket sales. Berikut adalah penjelasan langkah-langkahnya:

Kode `branch_income=smoker.groupby('Branch')['grossincome'].sum().sort_values(ascending=False)` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' dan menghitung jumlah total gross income (kolom 'gross income') untuk setiap cabang. Kita menggunakan metode `'groupby()'` untuk mengelompokkan data, memilih kolom 'gross income', dan menggunakan metode `'sum()'` untuk menjumlahkan nilai dalam setiap kelompok. Kemudian, hasilnya diurutkan secara menurun menggunakan metode `'sort_values()'`.

Kode `'print(branch_income)'` digunakan untuk mencetak hasil kelompokkan data berdasarkan cabang dan jumlah total gross income.

### 33. Total Gross Margin tiap Branch

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# kelompokkan data berdasarkan branch dan jumlah total gross income
branch_gross_margin = df.groupby('Branch')['gross margin percentage'].sum()

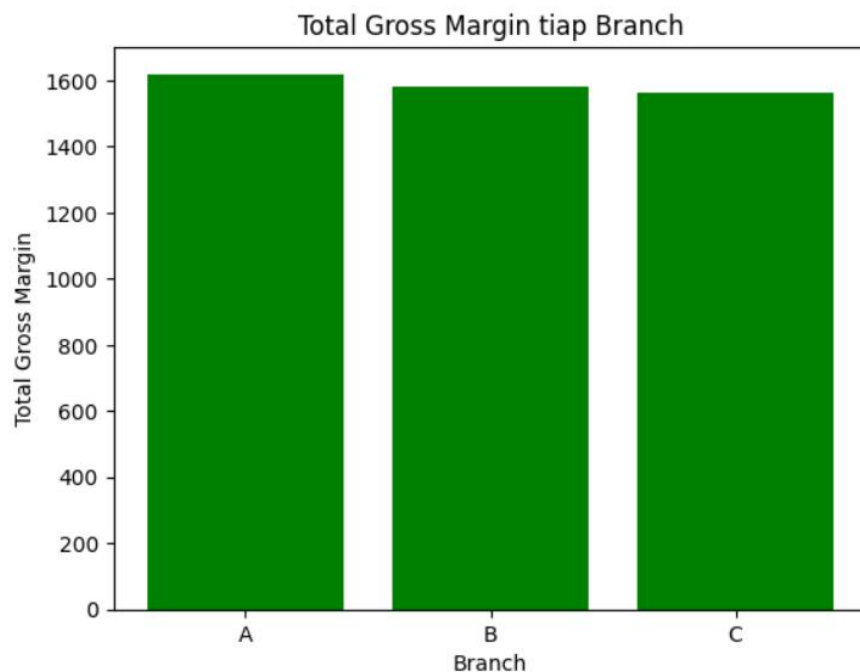
# hasil dalam bentuk tulisan
print(branch_gross_margin)

# bar chart
plt.bar(branch_gross_margin.index, branch_gross_margin.values, color='green')

# judul dan label chart
plt.title('Total Gross Margin tiap Branch')
plt.xlabel('Branch')
plt.ylabel('Total Gross Margin')

# tampilkan chart
plt.show()
```

```
Branch
A    1619.047619
B    1580.952381
C    1561.904762
Name: gross margin percentage, dtype: float64
```



Penjelasan:

Kode `branch_gross_margin = smoker.groupby('Branch')['gross margin percentage'].sum()` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' dan menghitung jumlah total gross margin percentage (kolom 'gross margin percentage') untuk setiap cabang. Kita menggunakan metode `groupby()` untuk mengelompokkan data, memilih kolom 'gross margin percentage', dan menggunakan metode `sum()` untuk menjumlahkan nilai dalam setiap kelompok.

Kode `print(branch_gross_margin)` digunakan untuk mencetak hasil kelompokkan data berdasarkan cabang dan jumlah total gross margin percentage.

Kode `plt.bar(branch_gross_margin.index, branch_gross_margin.values, color='green')` digunakan untuk membuat bar chart dari data hasil kelompokkan. Kita menggunakan `plt.bar()` dengan nilai cabang sebagai sumbu x dan nilai total gross margin percentage sebagai sumbu y. Parameter `color` digunakan untuk memberikan warna pada bar chart.

#### 34. Total Pembelian Tiap Branch

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# Hitung total pembelian tiap branch
total_purchase = smoker.groupby('Branch')['Total'].sum()

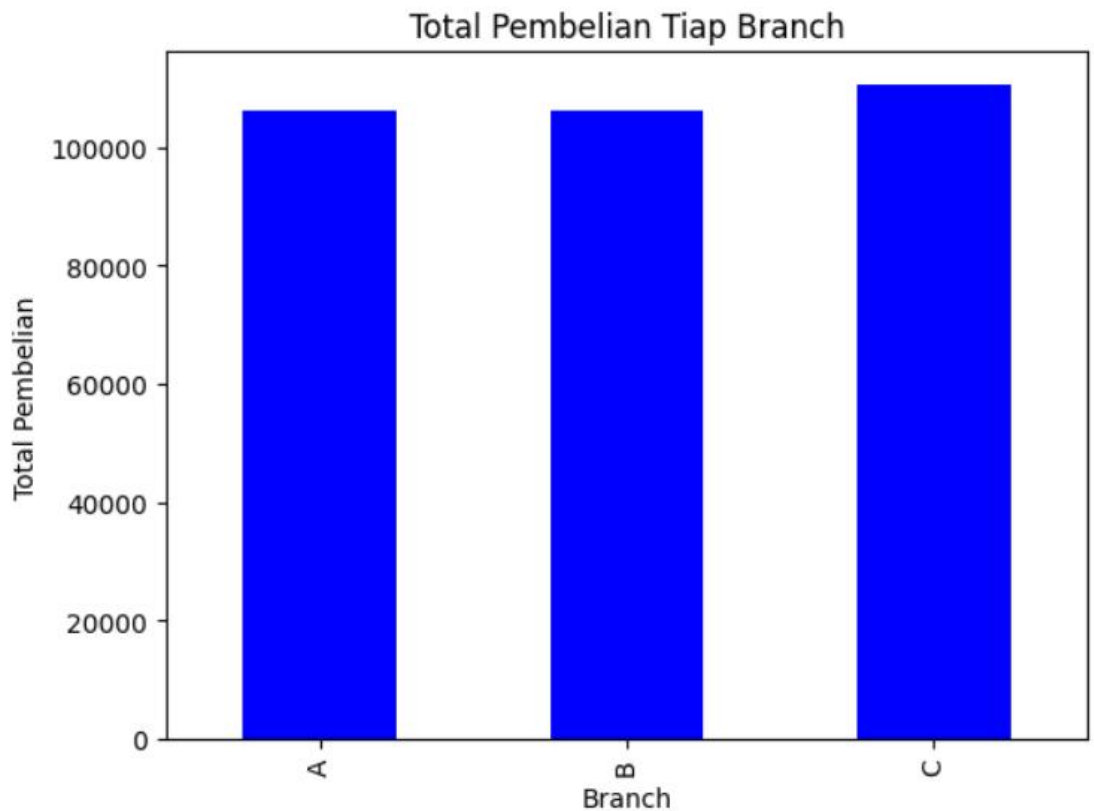
print(total_purchase)

# Buat stacked bar chart
total_purchase.plot(kind='bar', stacked=True, color='blue')

# Atur judul dan label sumbu
plt.title('Total Pembelian Tiap Branch')
plt.xlabel('Branch')
plt.ylabel('Total Pembelian')

# Tampilkan chart
plt.show()
plt.show()
```

```
Branch
A    106200.3705
B    106197.6720
C    110568.7065
Name: Total, dtype: float64
```



Penjelasan:

Kode `total_purchase = smoker.groupby('Branch')['Total'].sum()` digunakan untuk menghitung total pembelian (kolom 'Total') pada setiap cabang. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan cabang (kolom 'Branch') dan memilih kolom 'Total'. Selanjutnya, metode `sum()` digunakan untuk menjumlahkan nilai pembelian dalam setiap kelompok cabang.

Kode `print(total_purchase)` digunakan untuk mencetak hasil total pembelian tiap cabang.

Kode `total_purchase.plot(kind='bar', stacked=True, color='blue')` digunakan untuk membuat stacked bar chart dari data total pembelian per cabang. Kita menggunakan `plot(kind='bar')` untuk membuat bar chart, dan parameter `stacked=True` untuk menampilkan stacked chart. Parameter `color='blue'` digunakan untuk memberikan warna biru pada bar chart.



### 35. Total Tax per Branch

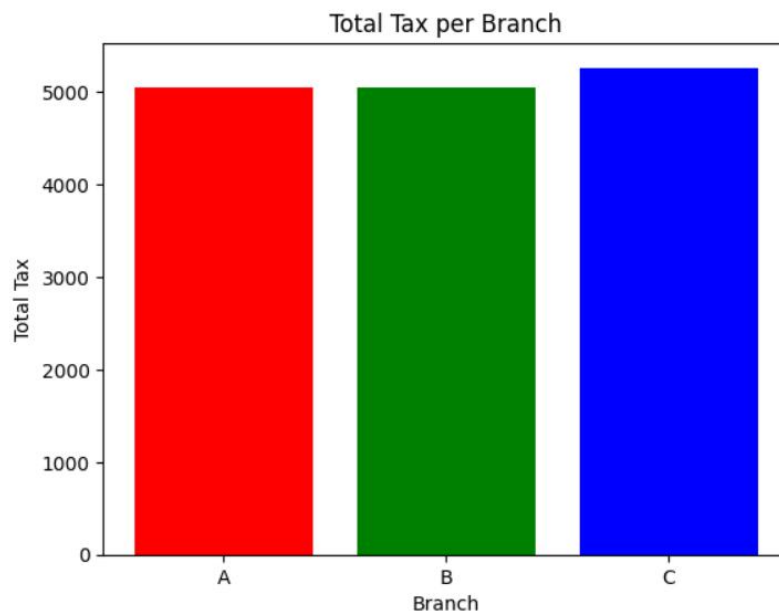
```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# hitung total tax dan kelompokkan berdasarkan branch
total_tax = df.groupby('Branch')['Tax 5%'].sum().reset_index()

print(total_tax)
# bar chart
plt.bar(total_tax['Branch'], total_tax['Tax 5%'], color=['red', 'green', 'blue'])
plt.title('Total Tax per Branch')
plt.xlabel('Branch')
plt.ylabel('Total Tax')
plt.show()
```

	Branch	Tax 5%
0	A	5057.1605
1	B	5057.0320
2	C	5265.1765



Penjelasan:

Kode `total_tax = df.groupby('Branch')['Tax 5%'].sum().reset_index()` digunakan untuk menghitung total pajak (kolom 'Tax 5%') pada setiap cabang. Kita menggunakan metode `groupby()` untuk mengelompokkan data berdasarkan cabang (kolom 'Branch') dan memilih kolom 'Tax 5%'. Selanjutnya, metode `sum()` digunakan untuk menjumlahkan nilai pajak dalam setiap kelompok cabang. Fungsi `reset_index()` digunakan untuk mengatur ulang indeks dari hasil pengelompokkan.

Kode `print(total\_tax)` digunakan untuk mencetak hasil total pajak tiap cabang.

### 36. Total Tipe Customer tiap Branch

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

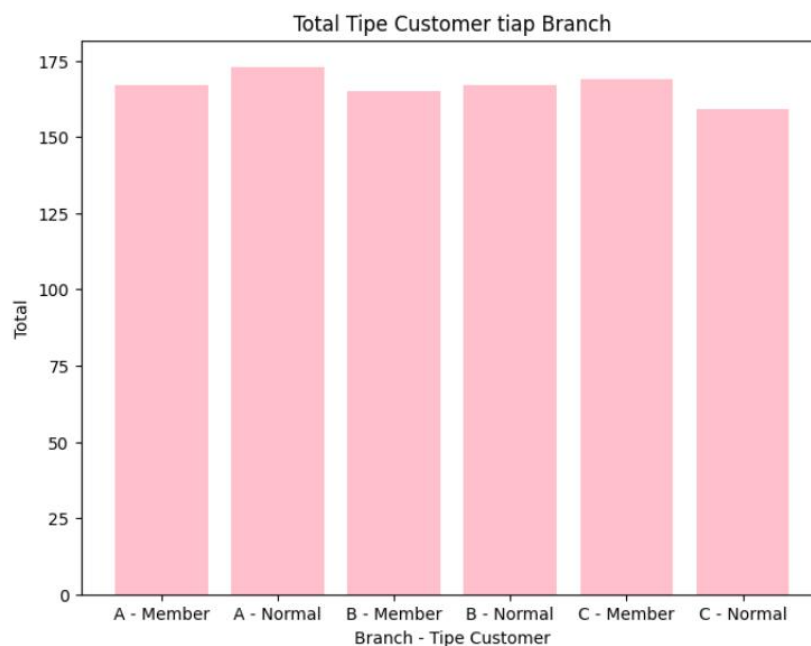
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# kelompokkan data berdasarkan branch dan customer type dan jumlah tiap customer
customer_type = df.groupby(["Branch", "Customer type"])["Invoice ID"].count().reset_index()
customer_type.columns = ["Branch", "Customer type", "Count"]

print(customer_type)

# bar chart
plt.figure(figsize=(8, 6))
plt.bar(customer_type["Branch"] + " - " + customer_type["Customer type"],
        customer_type["Count"], color='pink')
plt.title("Total Tipe Customer tiap Branch")
plt.xlabel("Branch - Tipe Customer")
plt.ylabel("Total")
plt.show()
```

	Branch	Customer type	Count
0	A	Member	167
1	A	Normal	173
2	B	Member	165
3	B	Normal	167
4	C	Member	169
5	C	Normal	159



Penjelasan:

Kode ``customer_type = smoker.groupby(["Branch", "Customer type"])["Invoice ID"].count().reset_index()`` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' (cabang) dan 'Customer type' (jenis pelanggan). Kemudian, kita menggunakan metode ``count()`` pada kolom 'Invoice ID' untuk menghitung jumlah pelanggan dalam setiap kelompok cabang dan jenis pelanggan. Fungsi ``reset_index()`` digunakan untuk mengatur ulang indeks dari hasil pengelompokkan.

Kode ``customer_type.columns = ["Branch", "Customer type", "Count"]`` digunakan untuk mengubah nama kolom hasil pengelompokkan menjadi "Branch", "Customer type", dan "Count". Hal ini dilakukan agar nama kolom menjadi lebih deskriptif dan mudah dibaca.

### 37. total mean rating per branch

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung mean rating per branch
mean_rating_per_branch = df.groupby('Branch')['Rating'].mean()

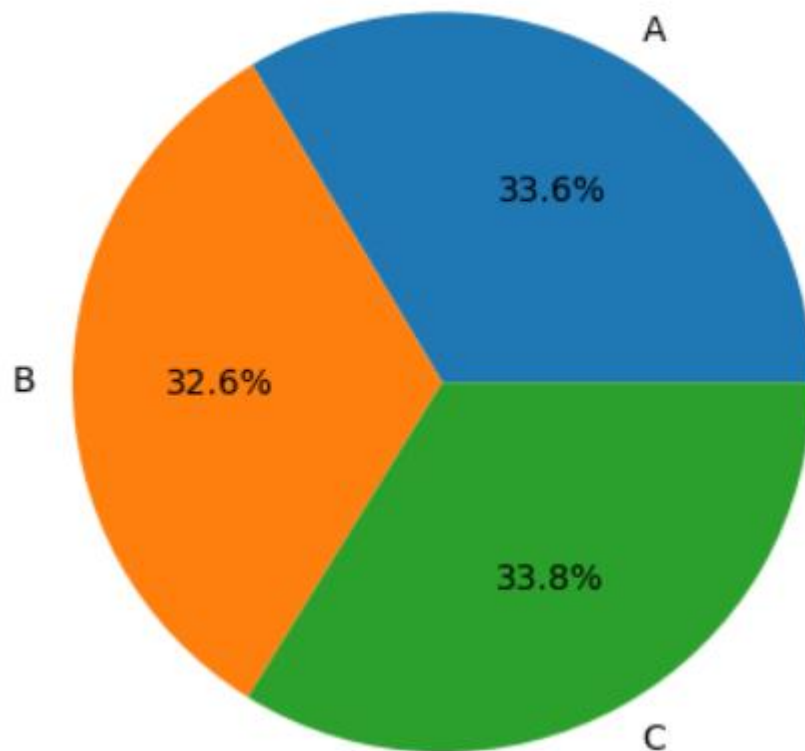
# Mengambil data branch dan mean rating
branches = mean_rating_per_branch.index
mean_ratings = mean_rating_per_branch.values

# Membuat pie chart
plt.pie(mean_ratings, labels=branches, autopct='%1.1f%%')

# Menentukan judul chart
plt.title('Mean Rating per Branch')

# Menampilkan pie chart
plt.show()
```

## Mean Rating per Branch



Penjelasan:

Kode `mean_rating_per_branch = df.groupby('Branch')['Rating'].mean()` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' (cabang) dan menghitung rata-rata dari kolom 'Rating' (rating). Fungsi `mean()` digunakan untuk menghitung rata-rata.

Kode `branches = mean_rating_per_branch.index` digunakan untuk mengambil indeks dari hasil pengelompokkan, yang merepresentasikan nilai cabang. Kode `mean_ratings = mean_rating_per_branch.values` digunakan untuk mengambil nilai rata-rata rating dari hasil pengelompokkan.

### 38. total payment per branch

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

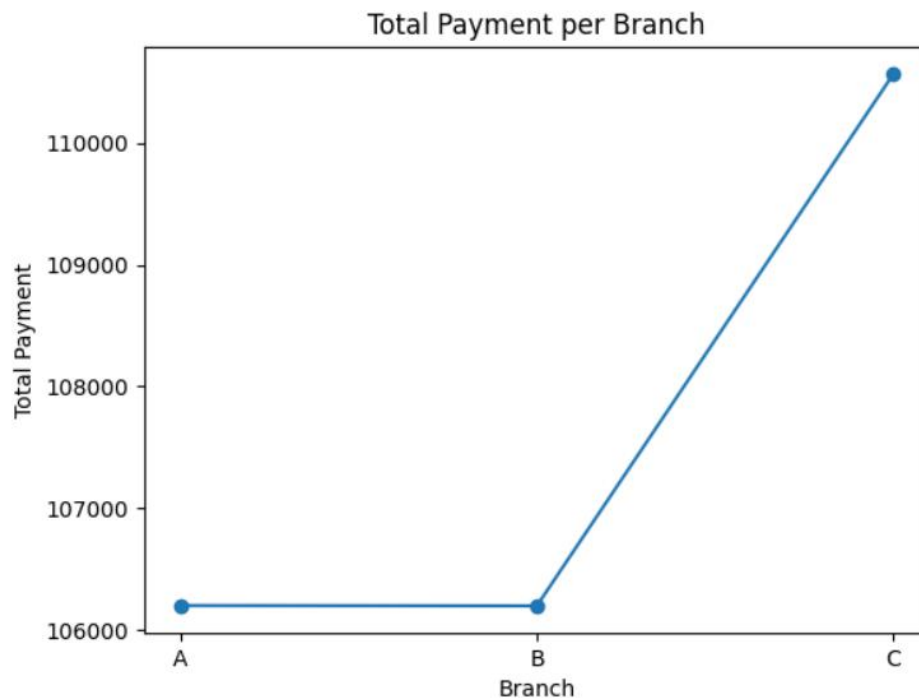
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales_csv.csv')

# Menghitung total payment per branch
```

```
total_payment_per_branch = df.groupby('Branch')['Total'].sum()

# Membuat line chart
plt.plot(total_payment_per_branch.index, total_payment_per_branch.values,
marker='o')
plt.xlabel('Branch')
plt.ylabel('Total Payment')
plt.title('Total Payment per Branch')

# Menampilkan line chart
plt.show()
```



Penjelasan:

Kode `total_payment_per_branch = df.groupby('Branch')['Total'].sum()` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' (cabang) dan menghitung jumlah total dari kolom 'Total' (total pembayaran). Fungsi `sum()` digunakan untuk menjumlahkan nilai-nilai tersebut. Dengan menjalankan kode di atas, variabel `total_payment_per_branch` akan berisi daftar total pembayaran per cabang.

### 39. Total Produk Tiap Branch

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

# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')
```

```
grouped_df = smoker.groupby(['Branch', 'Product line']).size().unstack()

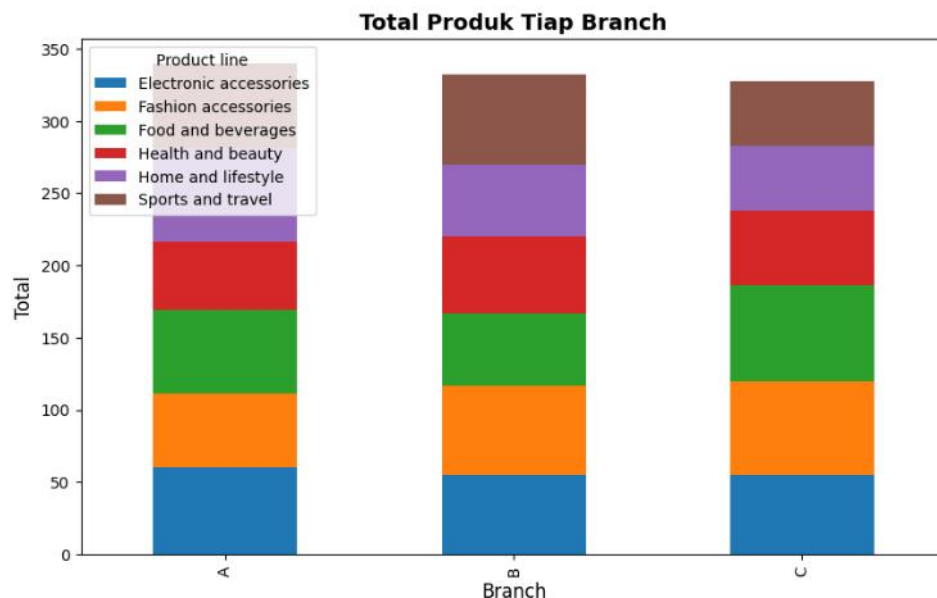
print(grouped_df)
# create a stacked bar chart
ax = grouped_df.plot(kind='bar', stacked=True, figsize=(10, 6))

# set the title and axis labels
ax.set_title('Total Produk Tiap Branch', fontsize=14, fontweight='bold')
ax.set_xlabel('Branch', fontsize=12)
ax.set_ylabel('Total', fontsize=12)

# show the plot
plt.show()
```

Product line \ Branch	Electronic accessories	Fashion accessories	Food and beverages
A	60	51	58
B	55	62	50
C	55	65	66

Product line \ Branch	Health and beauty	Home and lifestyle	Sports and travel
A	47	65	59
B	53	50	62
C	52	45	45



Penjelasan:

Kode ``grouped_df = smoker.groupby(['Branch', 'Product line']).size().unstack()`` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' (cabang) dan 'Product line' (jenis produk) dan menghitung jumlah produk yang dibeli menggunakan fungsi ``size()``. Kemudian, ``unstack()`` digunakan untuk mengubah hasil pengelompokkan menjadi bentuk yang lebih mudah diolah.

Kode `ax = grouped_df.plot(kind='bar', stacked=True, figsize=(10, 6))` digunakan untuk membuat stacked bar chart berdasarkan data yang telah dikelompokkan. Argumen `'kind='bar'` menentukan jenis chart yang ingin dibuat, yaitu bar chart. Argumen `'stacked=True'` digunakan untuk mengatur tumpukan (stacked) pada bar chart. Argumen `'figsize=(10, 6)'` menentukan ukuran gambar chart yang akan ditampilkan.

#### 40. Total Kota Tiap Branch

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

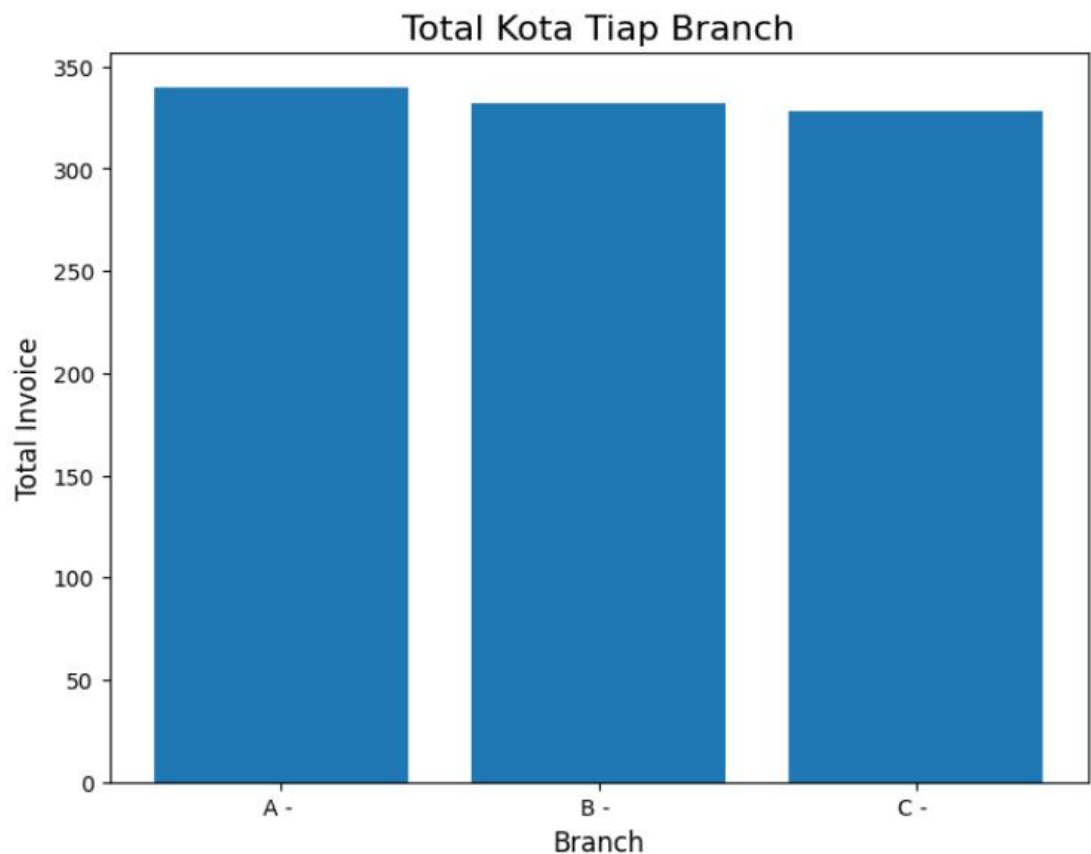
# Membaca dataset
df = pd.read_csv('Kelompok 1_supermarket sales.csv')

# group by branch and city
df_grouped = df.groupby(['Branch'])['Invoice ID'].count().reset_index()

print(df_grouped)

# plot the bar chart
plt.figure(figsize=(8,6))
plt.bar(df_grouped['Branch'] + ' - ', df_grouped['Invoice ID'])
plt.title('Total Kota Tiap Branch', fontsize=16)
plt.xlabel('Branch', fontsize=12)
plt.ylabel('Total Invoice', fontsize=12)
plt.show()
```

	Branch	Invoice ID
0	A	340
1	B	332
2	C	328



Penjelasan:

Kode di atas mengelompokkan data berdasarkan cabang (Branch) dan menghitung jumlah transaksi (Invoice ID) pada setiap cabang. Kemudian, hasil pengelompokkan tersebut ditampilkan. Berikut adalah penjelasan langkah-langkahnya:

Kode `df_grouped = smoker.groupby(['Branch'])['Invoice ID'].count().reset_index()` digunakan untuk mengelompokkan data berdasarkan kolom 'Branch' (cabang) dan menghitung jumlah transaksi dengan menggunakan kolom 'Invoice ID'. Fungsi `count()` digunakan untuk menghitung jumlah transaksi pada setiap cabang. Selanjutnya, `reset_index()` digunakan untuk mengatur ulang indeks dari hasil pengelompokkan.

Kode `print(df_grouped)` digunakan untuk menampilkan hasil pengelompokkan data, yaitu jumlah transaksi pada setiap cabang. Hasilnya akan berupa dataframe yang terdiri dari dua kolom, yaitu 'Branch' (cabang) dan 'Invoice ID' (jumlah transaksi).