

# **PENERAPAN K-MEANS++ CLUSTERING UNTUK SEGMENTASI CUSTOMER E-COMMERCE INDIA MENGUNAKAN METODE RFM (RECENCY FREQUENCY MONETARY)**

**DIMAS JABBAR ROSUL**

PROGRAM STUDI MATEMATIKA  
UNIVERSITAS INDONESIA



# Latar Belakang

Dalam penelitian kali ini, penulis akan menganalisis **data penjualan yang diperoleh dari e-commerce India**. Data-data yang diperoleh dari website e-commerce tersebut tentunya akan menghasilkan **insight/informasi yang berharga yang kemudian kita dapat menyimpulkan informasi yang didapat** serta merekomendasikan suatu keputusan bisnis sehingga pihak e-commerce dapat memaksimalkan penjualan mereka dengan baik.

Strategi bisnis yang digunakan adalah **Customer Relationship Management, yaitu teori yang membahas tentang hubungan perusahaan dengan pelanggan**, yang memiliki tujuan untuk meningkatkan hubungan dengan tiap pelanggan demi mencapai perkembangan perusahaan yang sehat. Sehingga untuk mengimplementasikan CRM, diperlukan analisis data berdasarkan RFM. **Analisis Recency, Frequency, Monetary (RFM) merupakan proses analisis perilaku pelanggan.**





# Analisis Recency, Frequency, Monetary (RFM)

## Recency

merupakan variabel untuk mengukur nilai pelanggan berdasarkan rentang waktu (tanggal, bulan, tahun) transaksi **terakhir pelanggan sampai saat ini.**



## Frequency

merupakan variabel untuk mengukur nilai pelanggan berdasarkan **jumlah transaksi yang dilakukan pelanggan dalam satu periode.** Semakin banyak jumlah transaksi yang dilakukan maka nilai semakin besar.



## Monetary

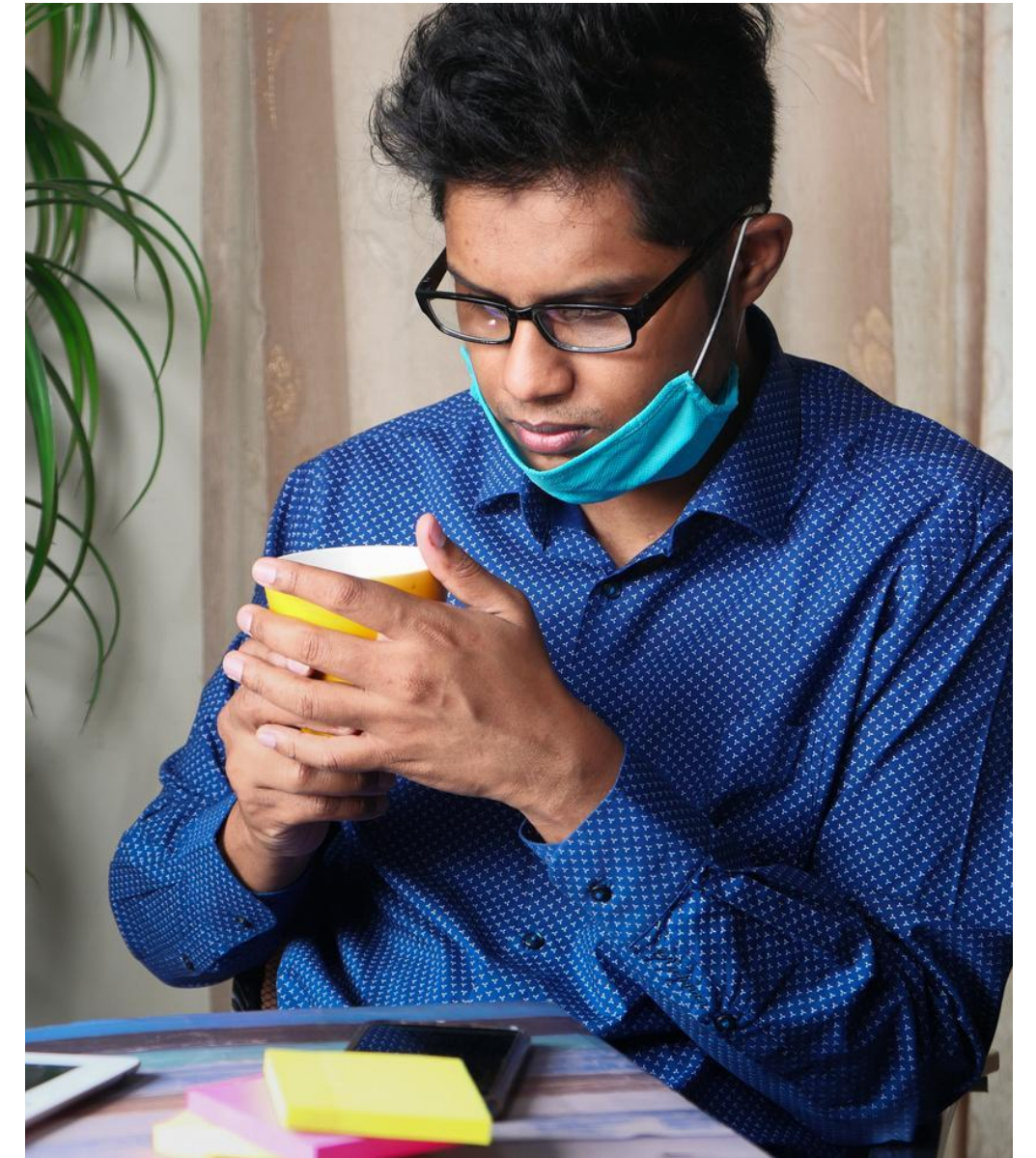
merupakan variabel untuk mengukur nilai pelanggan berdasarkan **jumlah besaran uang yang dikeluarkan pelanggan dalam satu periode.** Semakin banyak jumlah besaran uang yang dikeluarkan pelanggan maka nilai M semakin besar.



# Tujuan

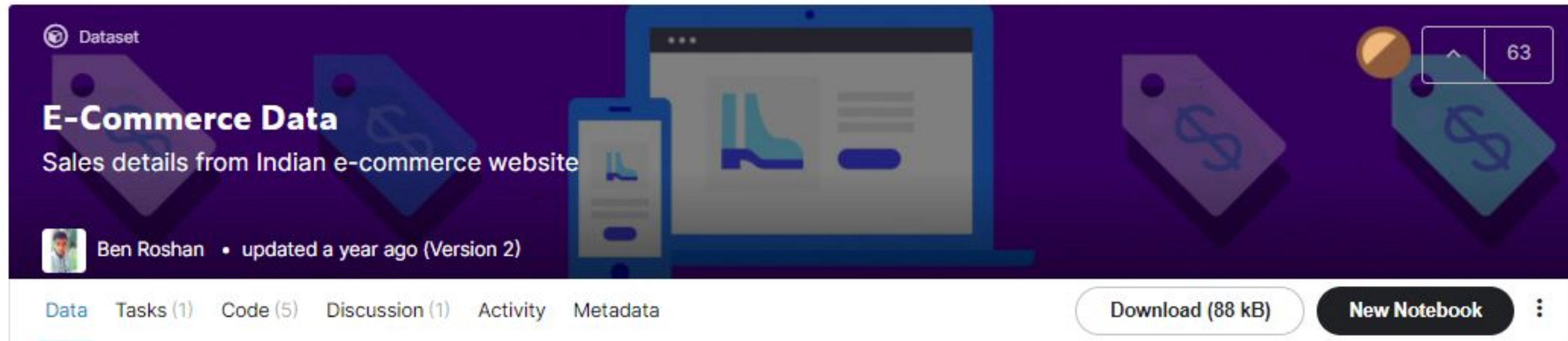
Pada project kali ini, penulis akan meng **klasterisasi pelanggan berdasarkan ciri-ciri pelanggan dengan RFM (Recency, Frequency, dan Monetary). Untuk menentukan strategi CRM yang tepat.**

Disamping itu, juga mengeksplorasi data dan menghasilkan visualisasi sehingga dari informasi tersebut, tim sales dan marketing dapat mengetahui problem dan **dapat mengevaluasi kinerja pemasaran dan penjualan.**



# Sumber Data

KAGGLE.COM



## List of Orders.csv

dataset tersebut mengandung informasi pelanggan, mencakup variabel data pelanggan seperti:

**Order ID:** ID pemesanan oleh pelanggan.

**Order Date:** tanggal pembelian.

**CustomerName:** nama pelanggan.

**State:** negara pelanggan.

**City:** kota pelanggan tinggal.

## Order Details.csv

dataset tersebut mengandung informasi secara detail mengenai pemesanan pada e-commerce tersebut, mencakup variabel seperti:

**Order ID:** pemesanan oleh pelanggan.

**Amount:** Harga barang yang telah dipesan

**Profit:** Keuntungan penjualan.

**Quantity:** Banyaknya pembelian barang tiap jenis item.

**Category:** Kategori barang.

**Sub-Category:** sub-category barang.



# Metode

Metode yang digunakan adalah menggunakan penerapan **K-Means++ Clustering** untuk segmentasi pelanggan berdasarkan prinsip **RFM (Recency, Frequency, Monetary)**. Penulis akan menggunakan tools google colab dengan **bahasa pemrograman python**.





# Pembahasan dan Hasil



## **Pre-processing**

penggabungan data dari berbagai sumber, pembersihan data



## **Visualisasi Data**

untuk mengeksplorasi data  
untuk melihat gambaran  
mengenai keseluruhan data



## **K-Means++**

segmentasi customer

Tahap 1

# Pre-Processing

code



# Membaca Data

Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import plotly.graph_objs as go
from sklearn.cluster import KMeans
import warnings
import os
```

```
from google.colab import drive
drive.mount('/gdrive')
```

```
import pandas as pd
List_of_Orders = pd.read_csv(r'/gdrive/MyDrive/KMMI/Project_KMMI/Indian_e_commerce/List of Orders.csv')
Order_Details = pd.read_csv(r'/gdrive/MyDrive/KMMI/Project_KMMI/Indian_e_commerce/Order Details.csv')
```

```
data1 = pd.merge(List_of_Orders, Order_Details, on='Order ID')
data1.head()
```

	Order ID	Order Date	CustomerName	State	City	Amount	Profit	Quantity	Category	Sub-Category
0	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	1275.0	-1148.0	7	Furniture	Bookcases
1	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	66.0	-12.0	5	Clothing	Stole
2	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	8.0	-2.0	3	Clothing	Hankerchief
3	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	80.0	-56.0	4	Electronics	Electronic Games
4	B-25602	01-04-2018	Pearl	Maharashtra	Pune	168.0	-111.0	2	Electronics	Phones

# Data cleansing

mengecek missing value dan Outliers.

Code

```
# Cetak jumlah missing value di setiap kolom
mv=data1.isna().sum()
print("\nJumlah missing value per kolom:\n", mv)
```

Jumlah missing value per kolom:

```
Order ID      0
Order Date    0
CustomerName  0
State         0
City          0
Amount        0
Profit        0
Quantity      0
Category      0
Sub-Category  0
dtype: int64
```

```
# Q1, Q3, dan IQR
Q1_Profit = data1['Profit'].quantile(0.25)
Q3_Profit = data1['Profit'].quantile(0.75)
IQR_Profit = Q3_Profit - Q1_Profit

Q1_Quantity = data1['Quantity'].quantile(0.25)
Q3_Quantity = data1['Quantity'].quantile(0.75)
IQR_Quantity = Q3_Quantity - Q1_Quantity

Q1_Amount = data1['Amount'].quantile(0.25)
Q3_Amount = data1['Amount'].quantile(0.75)
IQR_Amount = Q3_Amount - Q1_Amount

# Check ukuran (baris dan kolom) sebelum data yang outliers dibuang
print('Shape awal: ', data1.shape)

# Removing outliers
dataP = data1[~((data1['Profit'] < (Q1_Profit - 1.5 * IQR_Profit)) |(data1['Profit'] > (Q3_Profit + 1.5 * IQR_Profit)))]
dataQ = dataP[~((dataP['Quantity'] < (Q1_Quantity - 1.5 * IQR_Quantity)) |(dataP['Quantity'] > (Q3_Quantity + 1.5 * IQR_Quantity)))]
data = dataQ[~((dataQ['Amount'] < (Q1_Amount - 1.5 * IQR_Amount)) |(dataQ['Amount'] > (Q3_Amount + 1.5 * IQR_Amount)))]

# Check ukuran (baris dan kolom) setelah data yang outliers dibuang
print('Shape akhir dataP: ', dataP.shape)
print('Shape akhir dataQ : ', dataQ.shape)
print('Shape akhir data : ', data.shape)
```

```
Shape awal: (1500, 10)
Shape akhir dataP: (1209, 10)
Shape akhir dataQ : (1199, 10)
Shape akhir data : (1157, 10)
```



```
data['Profit'].describe()
```

```
count    1157.000000
mean      10.242005
std       32.537977
min      -79.000000
25%       -5.000000
50%        8.000000
75%       26.000000
max      108.000000
Name: Profit, dtype: float64
```

```
data['Profit Category']=pd.cut(data['Profit'],bins=[-79.000000,0.000000,8.000000,26.000000,108.000000],labels=['Rugi','Keuntungan Kecil','Keuntungan Sedang', 'Keuntungan Tinggi'])
```

```
data['Profit Category'].value_counts()
```

```
Rugi      402
Keuntungan Tinggi  282
Keuntungan Sedang  277
Keuntungan Kecil  194
Name: Profit Category, dtype: int64
```

```
data['Order Date'] = pd.to_datetime(data['Order Date'])
data['City'] = data['City'].astype('category')
data['CustomerName'] = data['CustomerName'].astype('category')
data['State'] = data['State'].astype('category')
data["Category"]=data["Category"].astype('category')
data["Sub-Category"]=data["Sub-Category"].astype('category')
data['Quantity']=data['Quantity'].astype('float64')
```

```
data['Month'] =pd.DataFrame(data["Order Date"])
data['Month'] =pd.DatetimeIndex(data['Order Date']).strftime('%m-%Y')
```

Order ID	object
Order Date	datetime64[ns]
CustomerName	category
State	category
City	category
Amount	float64
Profit	float64
Quantity	float64
Category	category
Sub-Category	category
Month	object
Day	object
Year	int64
Month_Number	int64
Profit Category	category
dtype:	object

# data.head()

melihat data teratas

```
Code
```

	Order ID	Order Date	CustomerName	State	City	Amount	Profit	Quantity	Category	Sub-Category	Month	Day	Year	Month_Number	Profit Category
1	B-25601	2018-01-04	Bharat	Gujarat	Ahmedabad	66.0	-12.0	5.0	Clothing	Stole	Jan	Thursday	2018	1	Rugi
2	B-25601	2018-01-04	Bharat	Gujarat	Ahmedabad	8.0	-2.0	3.0	Clothing	Hankerchief	Jan	Thursday	2018	1	Rugi
3	B-25601	2018-01-04	Bharat	Gujarat	Ahmedabad	80.0	-56.0	4.0	Electronics	Electronic Games	Jan	Thursday	2018	1	Rugi
8	B-25602	2018-01-04	Pearl	Maharashtra	Pune	119.0	-5.0	8.0	Clothing	Saree	Jan	Thursday	2018	1	Rugi
10	B-25603	2018-03-04	Jahan	Madhya Pradesh	Bhopal	24.0	-30.0	1.0	Furniture	Chairs	Mar	Sunday	2018	3	Rugi



Tahap 2

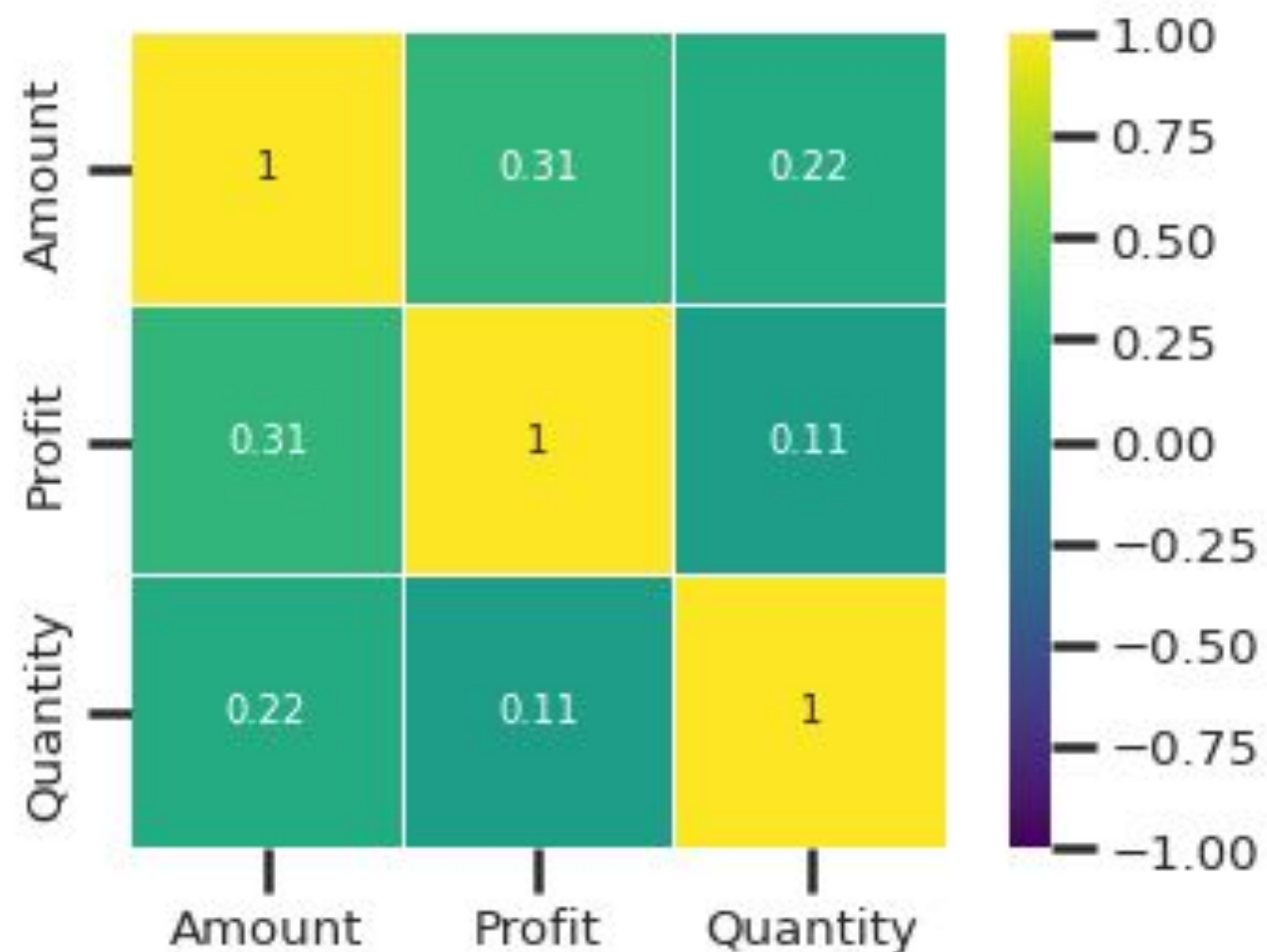
# Visualisasi

code

# Korelasi antar variabel

```
# Korelasi antar variabel data numerik
corr = data_num.corr()
c = sns.heatmap(corr, cmap='viridis', vmax=1, vmin= -1, linewidths=0.1, annot=True, square= True);
```

Code



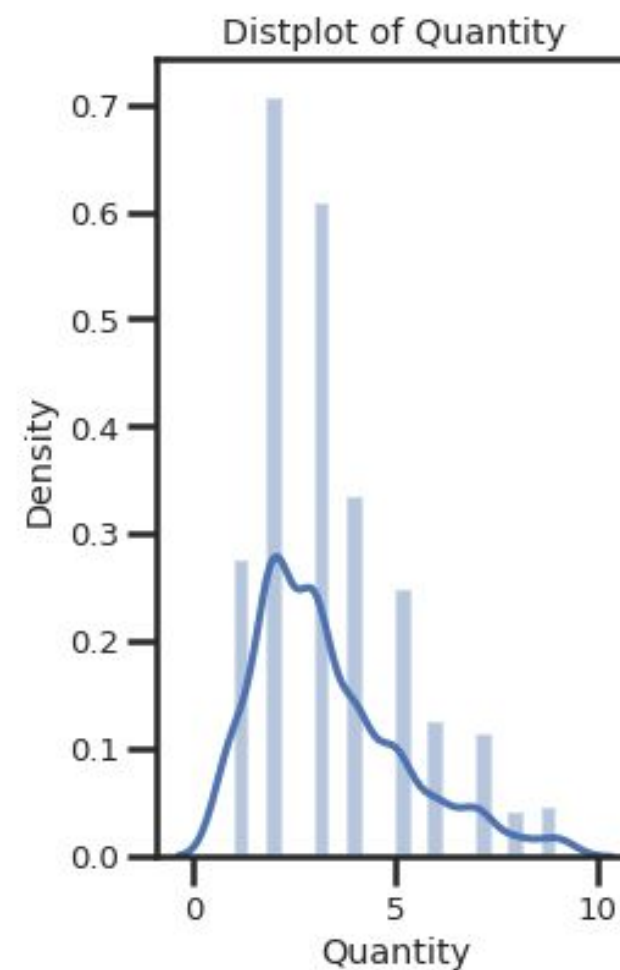
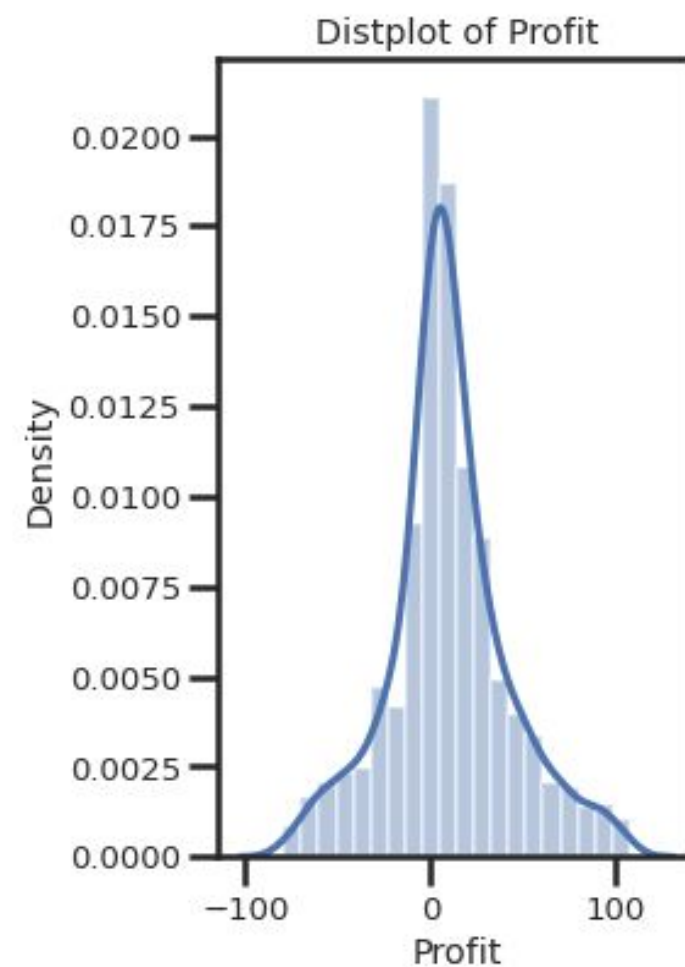
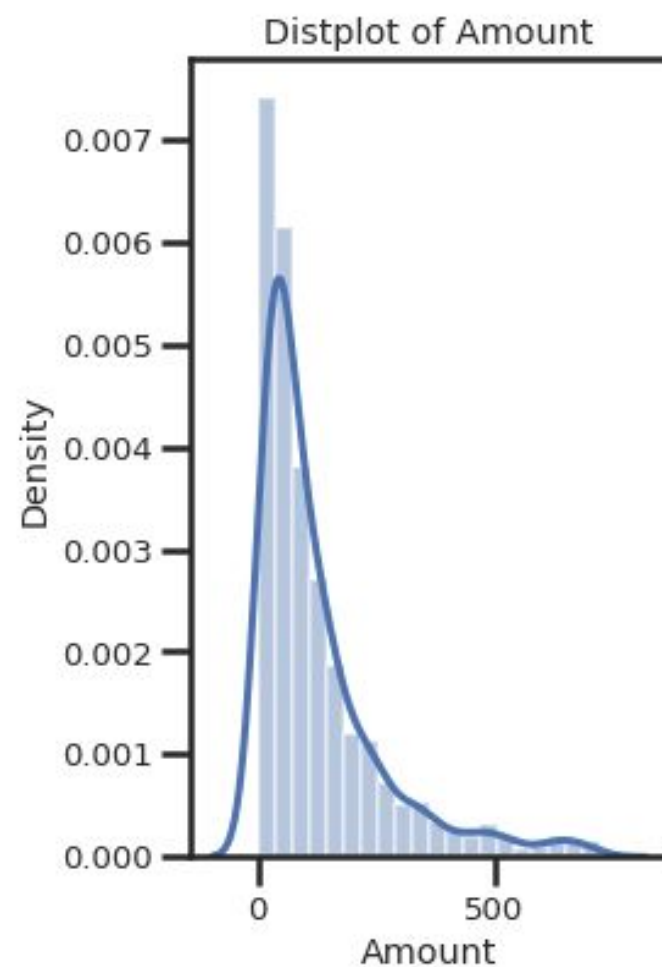
Dapat dilihat bahwa korelasi antar variabel agak kecil. Contoh nya pada variabel amount dan profit, korelasi nya hanya sekitar **0.31**. Menurut saya, hal ini agak menjanggal karena secara nalar jika jumlah uang yang dikeluarkan makin besar, maka semakin tinggi juga keuntungannya. Namun pada visualisasi tersebut tidak terlalu menunjukan korelasi yang kuat sehingga memerlukan analisa lebih lanjut.



# Distribusi Data

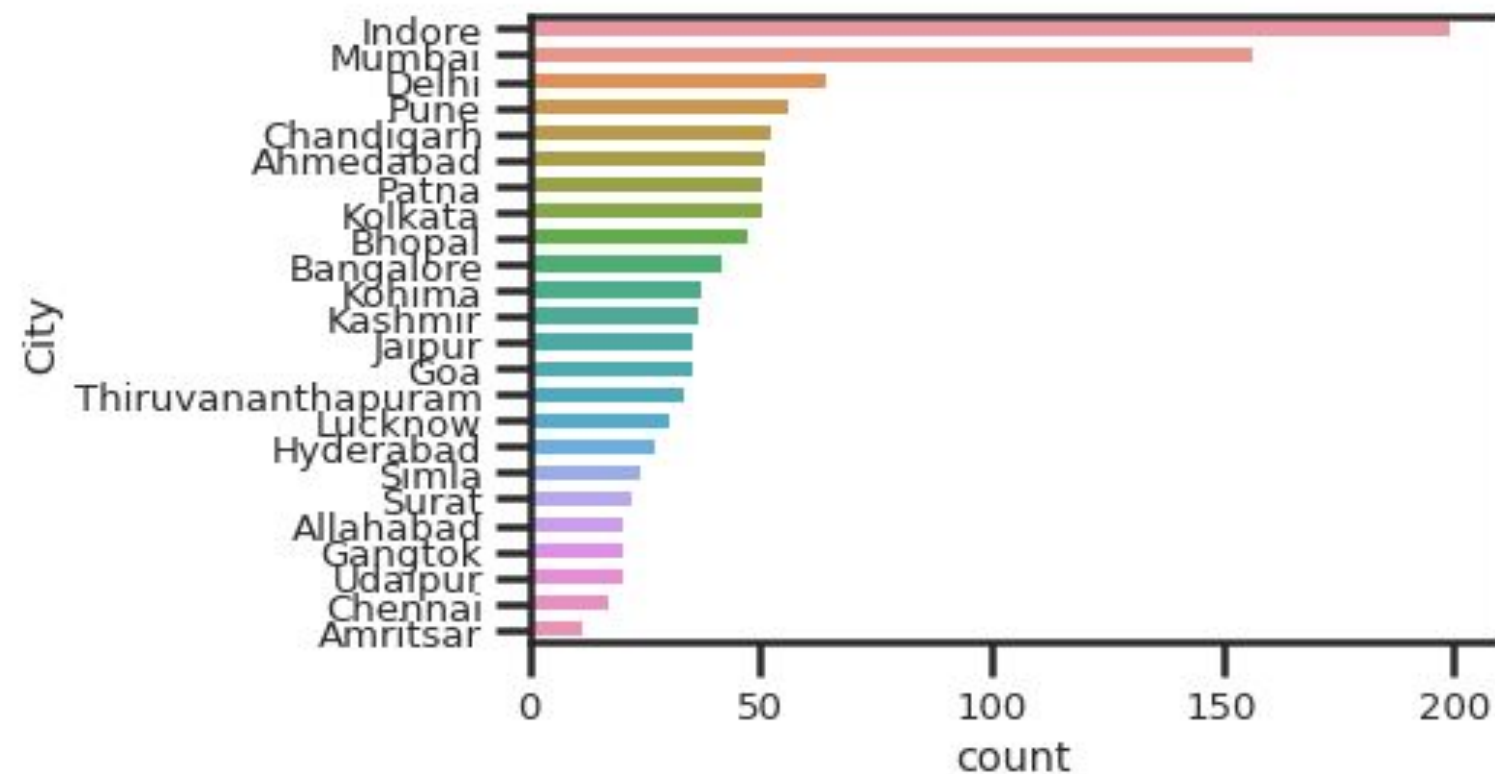
```
plt.figure(1 , figsize = (14 , 6))
n = 0
for x in ['Amount' , 'Profit' , 'Quantity']:
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.distplot(data[x] , bins = 20)
    plt.title('Distplot of {}'.format(x))
plt.show()
```

Code



jika diperhatikan rupanya pada distribusi data keuntungan kita dapat lihat bahwa terdapat sebagian besar yang mengalami kerugian ditandai dengan angka negatif sehingga hal ini harus diselidiki lebih lanjut.

```
# barplot kota yang melakukan pembelian
p = sns.countplot(y = "City", data = data ,order = data['City'].value_counts().index);
```

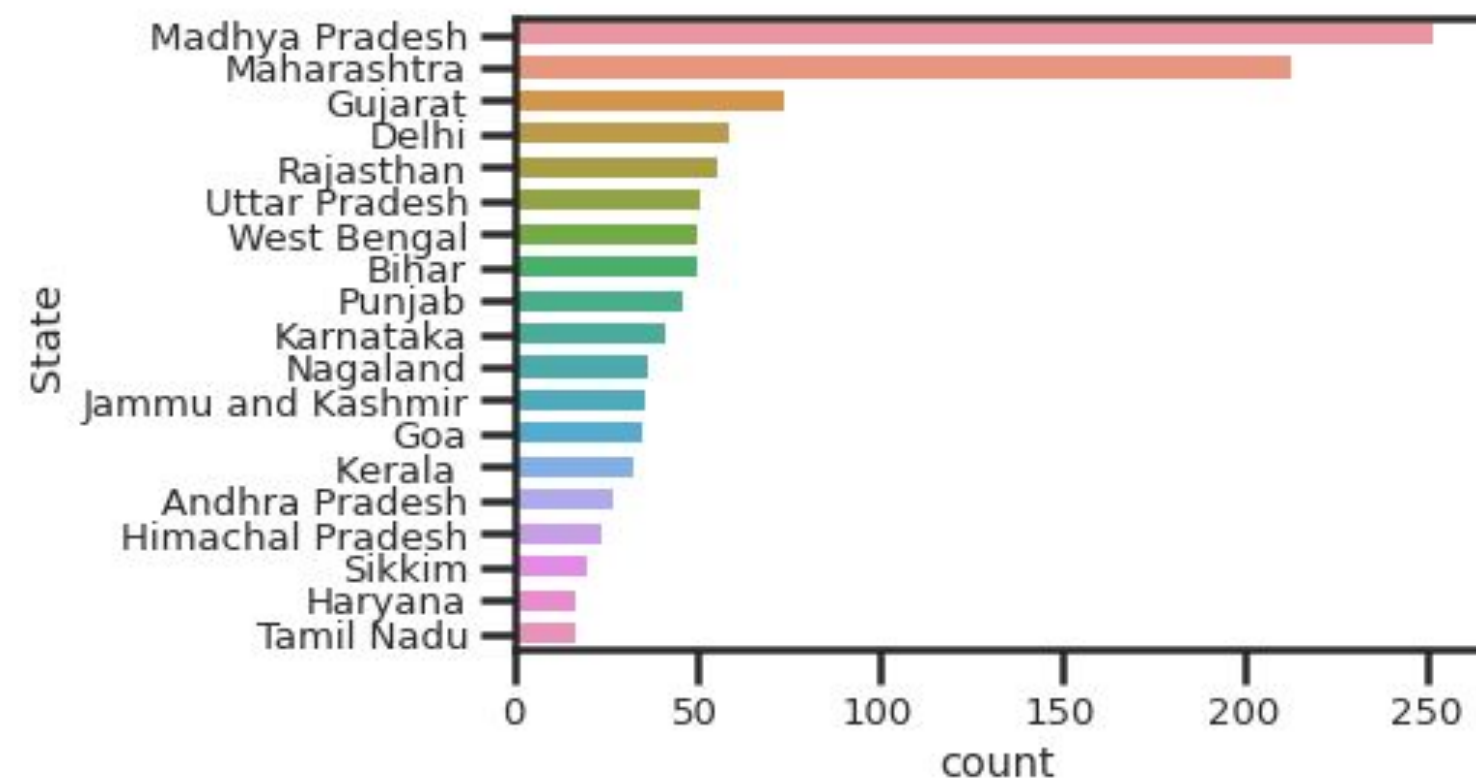


# Jumlah penjualan

Tiap negara bagian dan tiap kota

Code

```
# Barplot negara bagian yang melakukan pembelian
p = sns.countplot(y = "State", data = data ,order = data['State'].value_counts().index);
```



Pada kedua bar chart tersebut dapat dilihat bahwa daerah yang paling banyak membeli produk pada e-commerce ini adalah negara Madhya Pradesh pada Kota Indore dan negara bagian maharashtra pada kota mumbai sehingga saya merekomendasikan bagi tim marketing untuk mengekspansi negara bagian dan kota lainnya sehingga pemasaran semakin meluas.



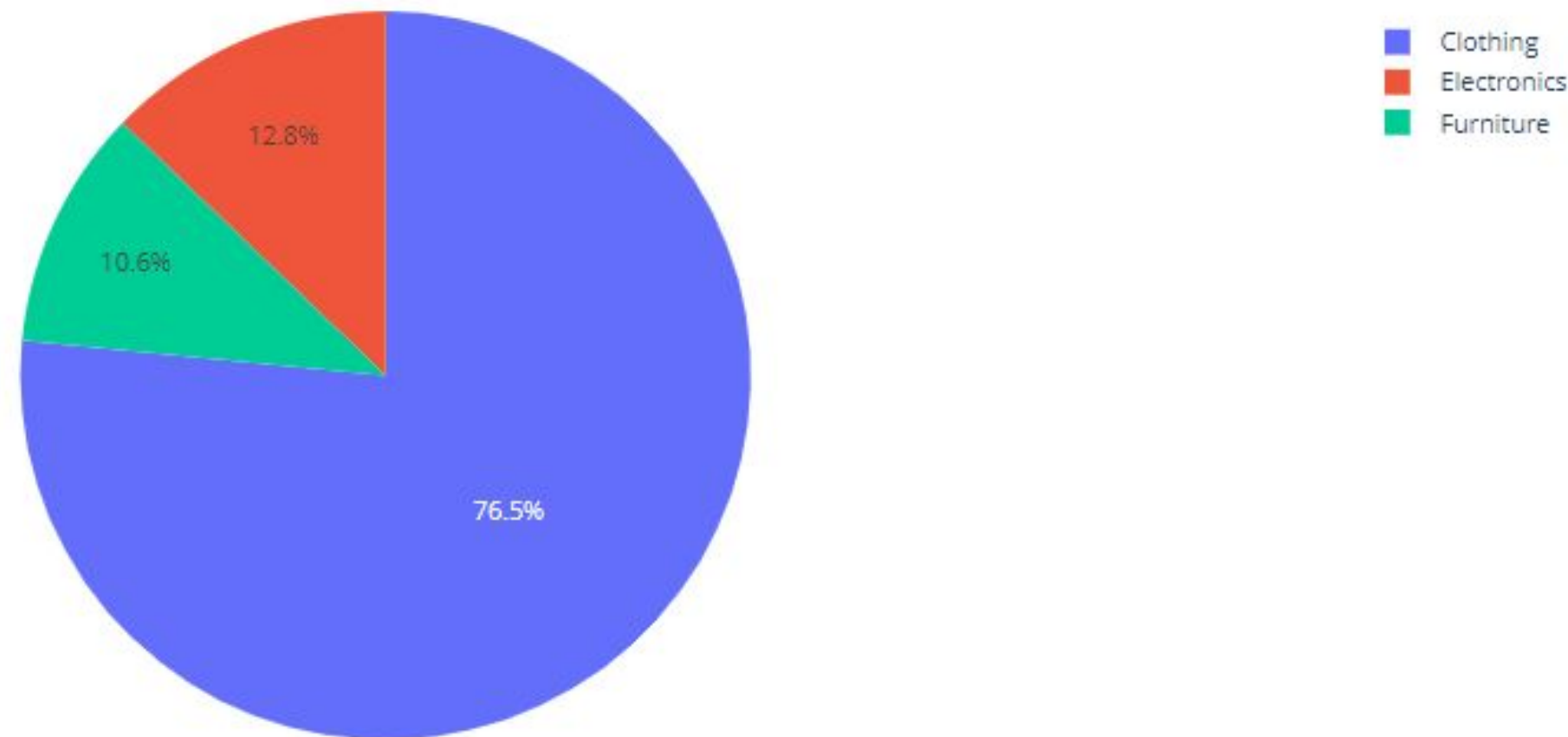
# Proporsi Jumlah penjualan

Tiap negara kategori dan subkategori

Code

```
details_category = data.groupby('Category').sum().reset_index()
fig = px.pie(details_category, values='Quantity', names='Category', color='Category',
            color_discrete_map={'Clothing':'cyan',
                                'Electronics':'royalblue',
                                'Furniture':'darkblue'},
            title='Total Quantity Sold per Category')
fig.show()
```

Total Quantity Sold per Category



Dari Pie chart dapat kita lihat bahwa penjualan terbanyak terjadi pada produk pada kategori clothing / pakaian. Sementara produk elektronik dan furniture hampir sama besar. Kemudian kita melihat jumlah penjualan per sub-category.

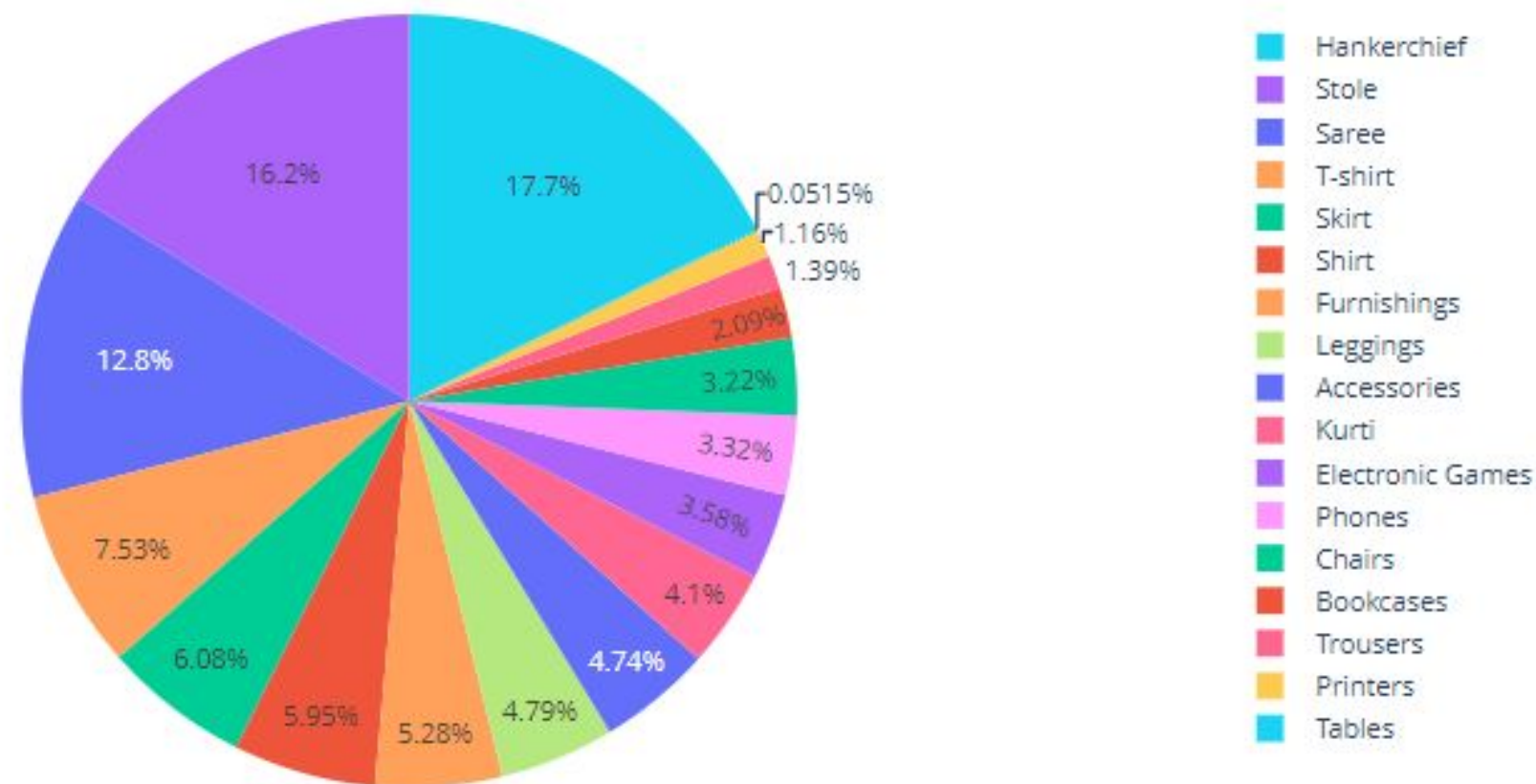
# Proporsi Jumlah penjualan

Tiap negara kategori dan subkategori

Code

```
details_subcategory = data.groupby('Sub-Category').sum().reset_index()
fig = px.pie(details_subcategory, values='Quantity', names='Sub-Category', color='Sub-Category',
             title='Total Quantity Sold per Sub-Category')
fig.show()
```

Total Quantity Sold per Sub-Category

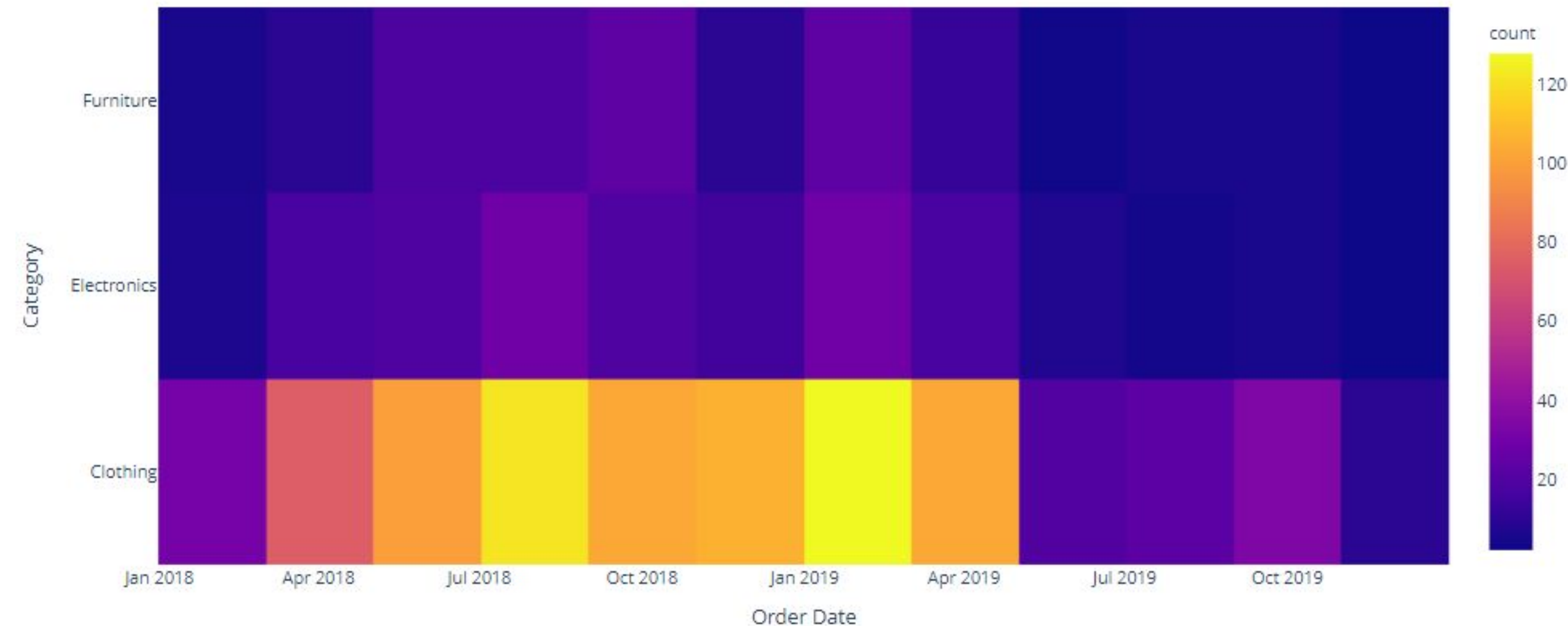


Berdasarkan sub-kategori barang yang terjual, penjualan terbanyak ada pada handkerchief, disusul stole, saree lalu dilanjut oleh jenis barang lainnya. Untuk saat ini, rencana awal yang mungkin akan direkomendasikan adalah melakukan promosi pada ketiga produk dengan penjualan terbanyak namun, kebijakan ini nantinya akan disesuaikan lagi dengan analisis lebih lanjut.

# Penjualan barang

dari waktu ke waktu tiap kategori dan subkategori

```
fig = px.density_heatmap(data, x="Order Date", y="Category")  
fig.show()
```



Code

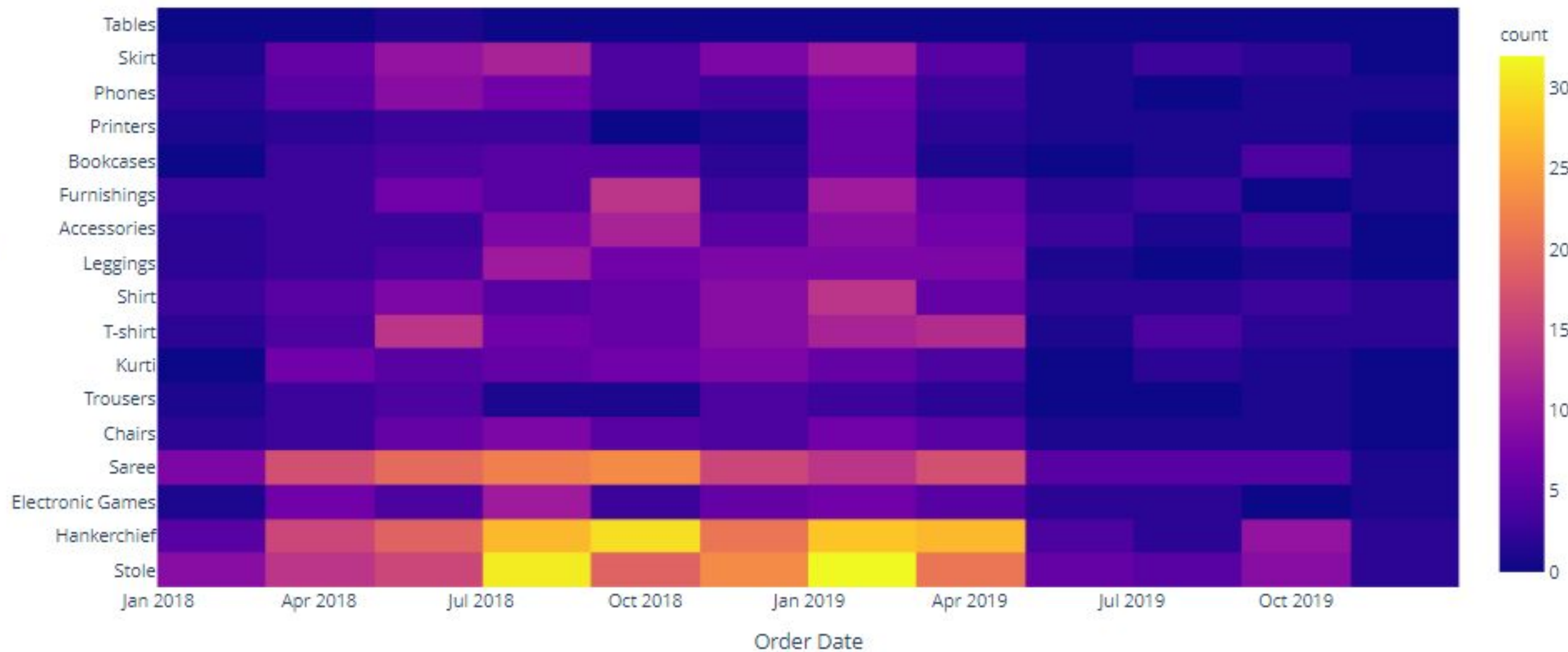
Berdasarkan waktu ke waktu, ternyata clothing juga menunjukkan performa yang lebih baik dari segi penjualan. Dari januari 2018 hingga juli 2018 mengalami peningkatan penjualan barang, fluktuatif cenderung stabil hingga april 2019 namun mengalami penurunan drastis pada bulan selanjutnya. Begitu Pula dengan produk lainnya bahwa setelah april 2019 produk-produk tersebut mengalami penurunan penjualan, sehingga harus diadakannya evaluasi mengenai pemasaran pada tanggal tersebut.



# Penjualan barang

dari waktu ke waktu tiap kategori dan subkategori

Code

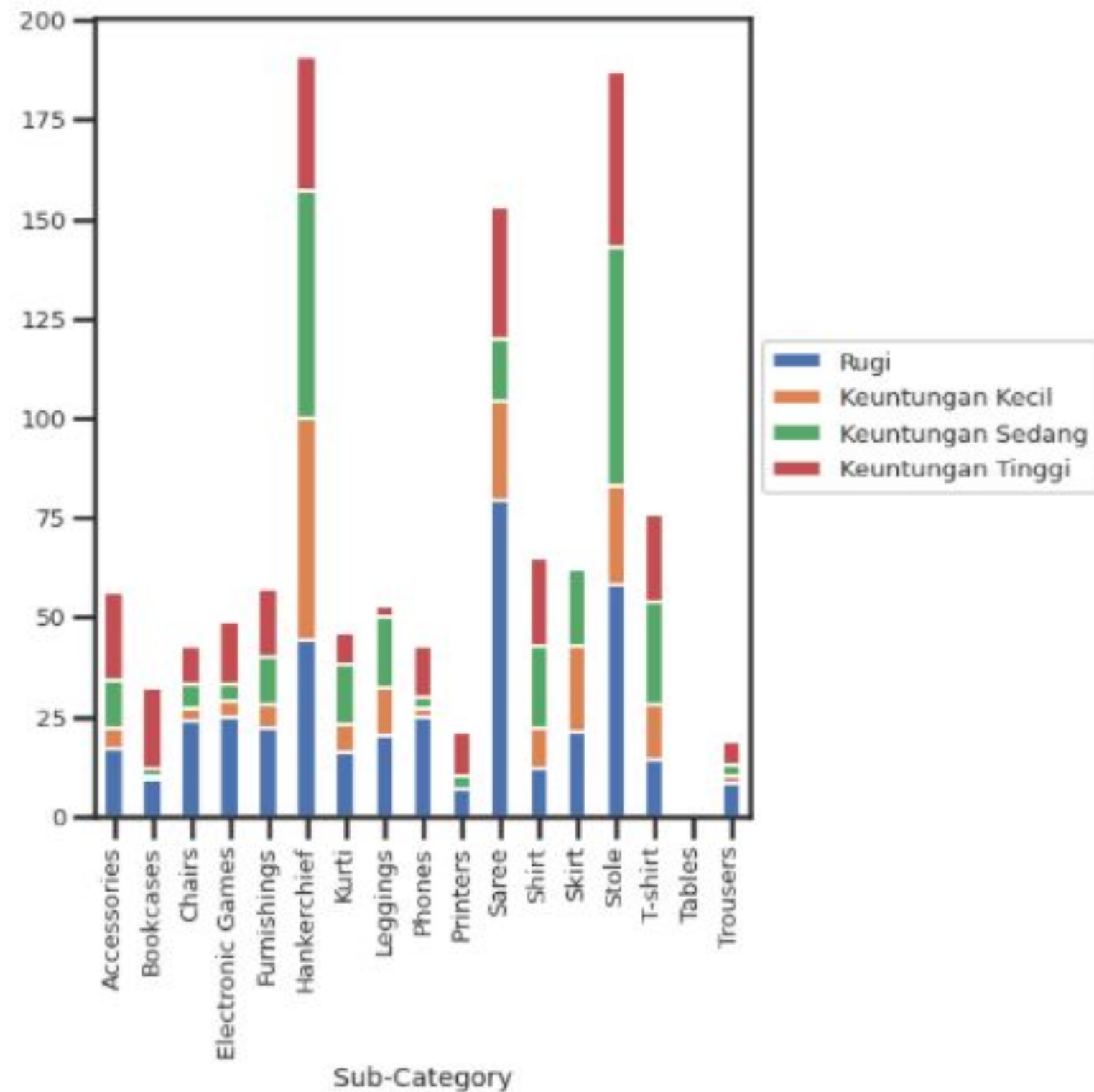


Dari diagram heatmap berdasarkan sub-category dapat dilihat juga bahwa dari waktu ke waktu bahwa penjualan handkerchief, stole, dan saree menunjukkan adanya peningkatan penjualan fluktuatif cenderung stabil hingga april 2019, sementara untuk penjualan produk lainnya terjual dengan jumlah secara merata namun tidak sebanyak 3 produk tadi.

# Proporsi keuntungan dan kerugian penjualan

berdasarkan tiap sub kategori barang

Code



```
#Stacked Chart untuk kategori Profit Total Perbelanjaan tiap negara
CT = pd.crosstab(index=data["Sub-Category"], columns=data["Profit Category"])
p = CT.plot(kind="bar", figsize=(8,8), stacked=True)
ax = plt.subplot(111)

# Shrink current axis by 20%
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.8, box.height])

# Put a legend to the right of the current axis
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
```



# Jumlah keuntungan dan kerugian penjualan

berdasarkan tiap sub kategori barang

Code

```
profit_by_Sub_Category = data.groupby('Sub-Category').sum().reset_index().sort_values(['Profit'])
profit_by_Sub_Category["Color"] = np.where(profit_by_Sub_Category["Profit"]<0, 'Loss', 'Profit')

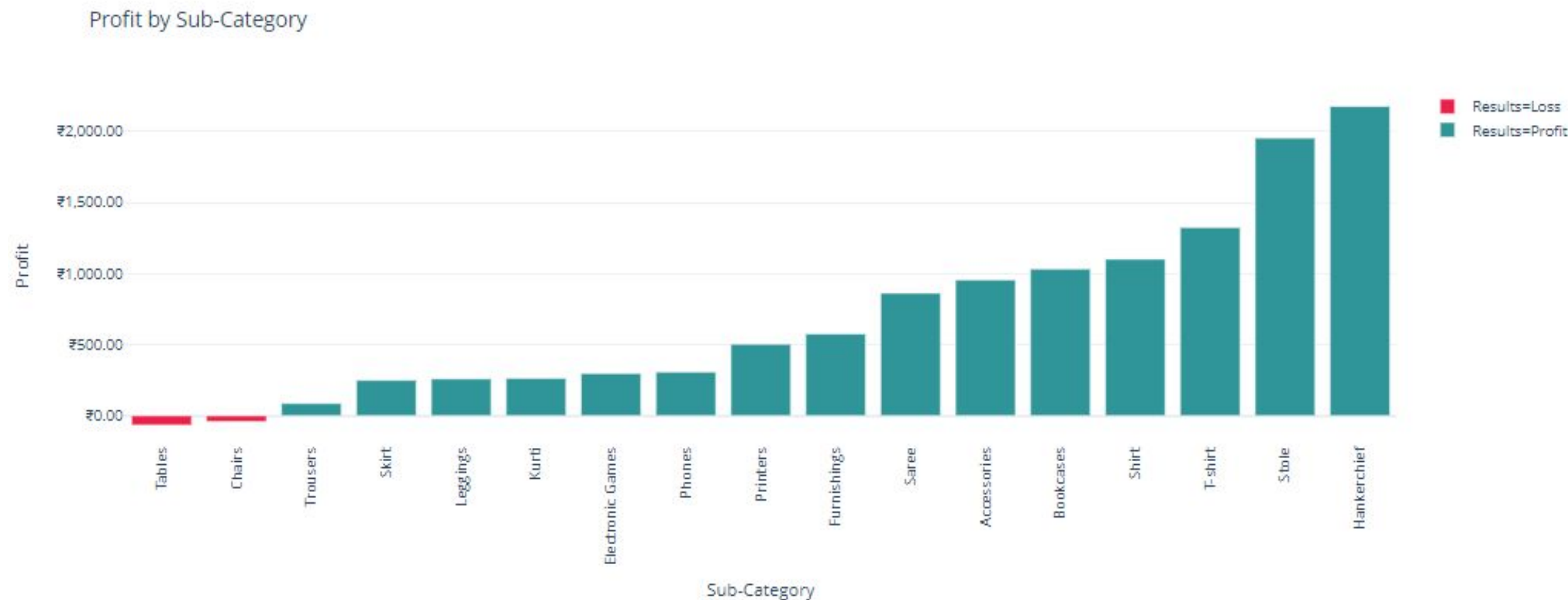
fig = px.bar(profit_by_Sub_Category, x='Sub-Category', y='Profit',
             color='Color', color_discrete_map={
                 'Loss': '#EC2049',
                 'Profit': '#2F9599'},
             title="Profit by Sub-Category",
             labels=dict(Color="Results"),
             template='plotly_white')

# Disabling Zoom
fig.layout.xaxis.fixedrange = True
fig.layout.yaxis.fixedrange = True

fig.update_layout(yaxis_tickprefix = '₹', yaxis_tickformat = ',.2f')

fig.update_xaxes(
    tickangle = -90,
    title_text = "Sub-Category",
)

fig.show()
```



Pada diagram batang tersebut dapat dilihat bahwa produk dengan keuntungan tertinggi adalah handkerchief, saree, dan T-shirt. Jika diperhatikan lagi pada visualisasi sebelumnya bahwa penjualan tertinggi adalah pada produk handkerchief, saree dan stole. Namun pada diagram ini dapat dilihat seksama bahwa T-shirt cukup menghasilkan keuntungan yang baik walaupun jumlah penjualannya kurang dari stole. Kemudian ada beberapa kerugian pada table dan chairs sehingga saya tidak merekomendasikan bagi tim bisnis untuk memberi diskon pada produk-produk tersebut, begitu pula dengan produk trousers, skirt, legging, kurti, electronic games, phones untuk saat ini.

```
profit_by_state = data.groupby('State').sum().reset_index().sort_values(['Profit'])
profit_by_state["Color"] = np.where(profit_by_state["Profit"]<0, 'Loss', 'Profit')
```

```
fig = px.bar(profit_by_state, x='State', y='Profit',
             color='Color', color_discrete_map={
                 'Loss': '#EC2049',
                 'Profit': '#2F9599'},
             title="Profit by State",
             labels=dict(Color="Results"),
             template='plotly_white')
```

```
fig.update_layout(yaxis_tickprefix = '₹', yaxis_tickformat = ',.2f')
```

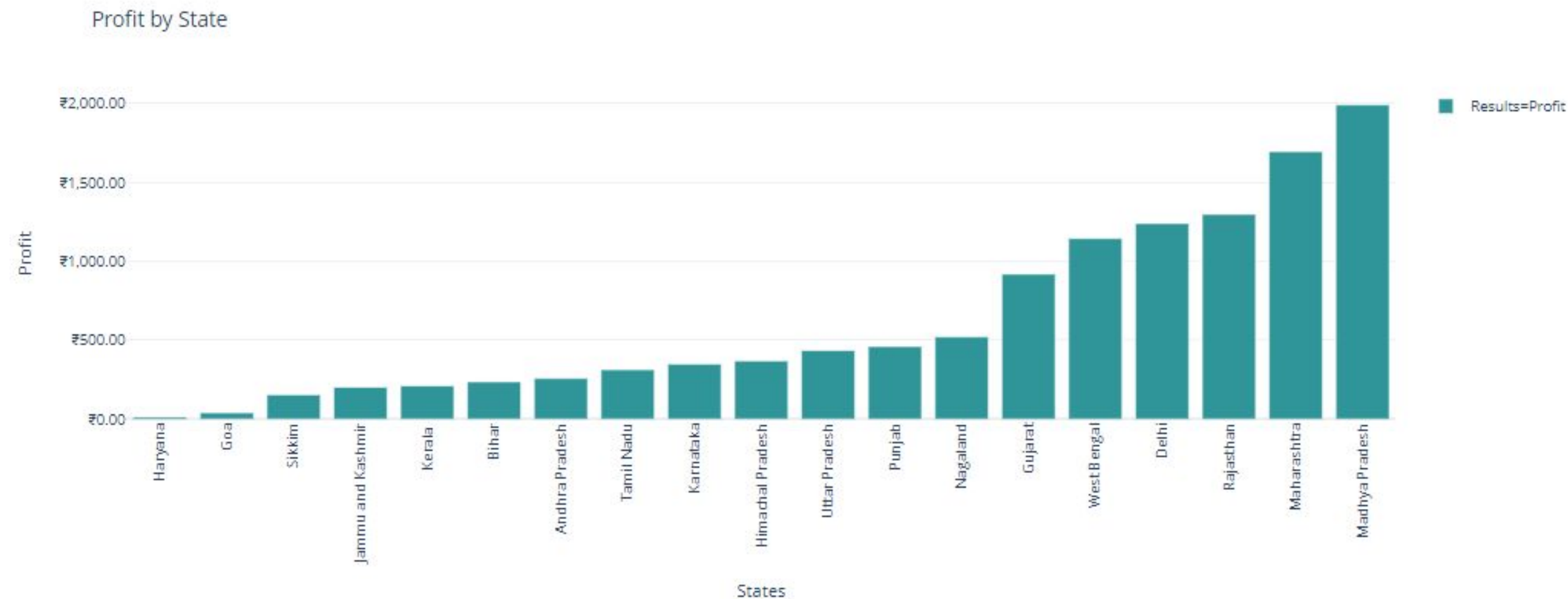
```
fig.update_xaxes(
    tickangle = -90,
    title_text = "States",
)
```

```
fig.show()
```

# Jumlah keuntungan dan kerugian penjualan

berdasarkan tiap negara bagian

Code



Keuntungan tertinggi yang didapat berasal dari negara bagian madhya pradesh, maharashtra, rajasthan, delhi, west bengal, dan gujarat. Sehingga saya merekomendasikan untuk meningkatkan performa pemasaran pada negara bagian tersebut serta mengevaluasi penjualan pada negara bagian yang lainnya terutama pada negara bagian Haryana dan goa karena keuntungan nya sangat kecil.



Tahap 3

# K-Means++

code

# Menyiapkan Tabel RFM

```
#Menyiapkan Tabel RFM
data_rfm = data[['Order ID','Order Date','CustomerName','Amount']]
data_rfm.head()
```

	Order ID	Order Date	CustomerName	Amount
1	B-25601	2018-01-04	Bharat	66.0
2	B-25601	2018-01-04	Bharat	8.0
3	B-25601	2018-01-04	Bharat	80.0
8	B-25602	2018-01-04	Pearl	119.0
10	B-25603	2018-03-04	Jahan	24.0

```
#melihat tanggal terbaru dan terlama
data_rfm['Order Date'].min(),data_rfm['Order Date'].max()

(Timestamp('2018-01-04 00:00:00'), Timestamp('2019-12-03 00:00:00'))
```

Code

```
import datetime as dt
NOW = dt.datetime(2019, 12, 3)
```

```
#Tabel RFM
rfm= data_rfm.groupby('CustomerName').agg({'Order Date': lambda date: (NOW - date.max()).days, #Recency
                                           'Order ID': 'count', #Frequency
                                           'Amount': 'sum'}) #Monetary value
rfm.head()
```

	Order Date	Order ID	Amount
CustomerName			
Aakanksha	695	2	74.0
Aarushi	256	8	2029.0
Aashna	600	6	1576.0
Aastha	403	4	259.0
Aayush	383	5	340.0



# Menyiapkan Tabel RFM

Code

```
#mengganti nama kolom
rfm.columns=['recency','frequency','monetary']
rfm['recency'] = rfm['recency'].astype(int)
rfm.head()
```

	recency	frequency	monetary
CustomerName			
Aakanksha	695	2	74.0
Aarushi	256	8	2029.0
Aashna	600	6	1576.0
Aastha	403	4	259.0
Aayush	383	5	340.0

```
#STANDARISASI
from sklearn.preprocessing import StandardScaler

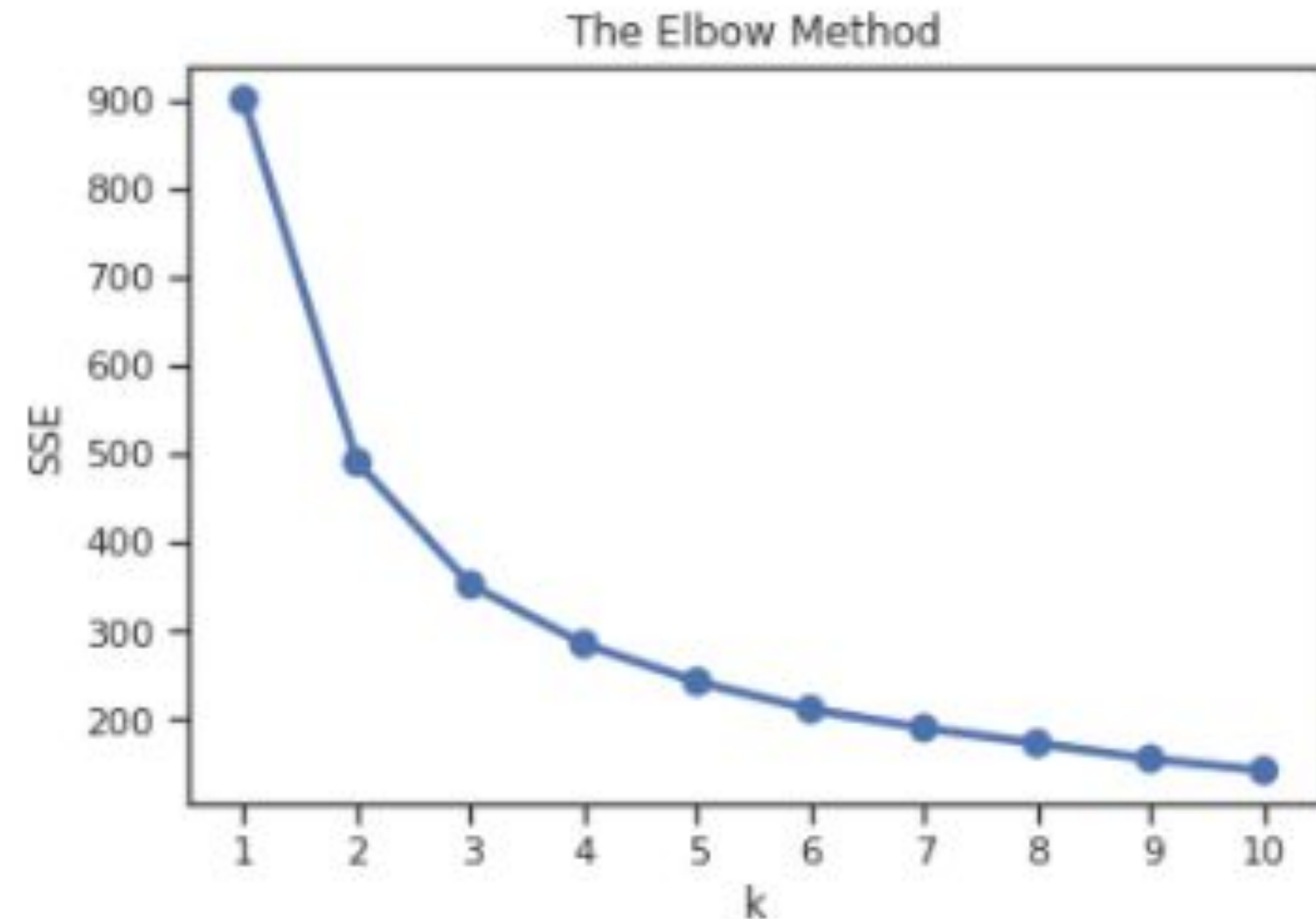
scale = StandardScaler()
scaleddata = pd.DataFrame(scale.fit_transform(rfm), columns=rfm.columns)
scaleddata.head()
```

	recency	frequency	monetary
0	2.206822	-0.601030	-0.839629
1	-0.771122	1.341258	3.125671
2	1.562392	0.693828	2.206857
3	0.226048	0.046399	-0.464396
4	0.090378	0.370114	-0.300105

# Menentukan nilai K

Code

```
#ELBOW METHOD
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
sse = {}
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(scaleddata)
    sse[k] = kmeans.inertia_ # SSE to closest cluster centroid
plt.title('The Elbow Method')
plt.xlabel('k')
plt.ylabel('SSE')
sns.pointplot(x=list(sse.keys()), y=list(sse.values()))
plt.show()
```



# K-Means++

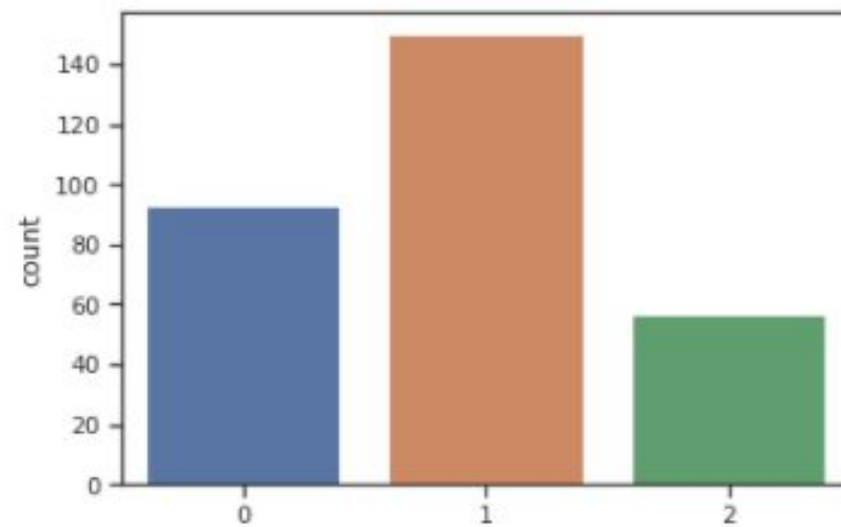
Code

```
#K-means
k=3
kmPP = cluster.KMeans(n_clusters=k, init='k-means++', max_iter=300, tol=0.0001, n_jobs=-1, random_state = random_state)
kmPP.fit(scaleddata)
C_kmpp = kmPP.predict(scaleddata)

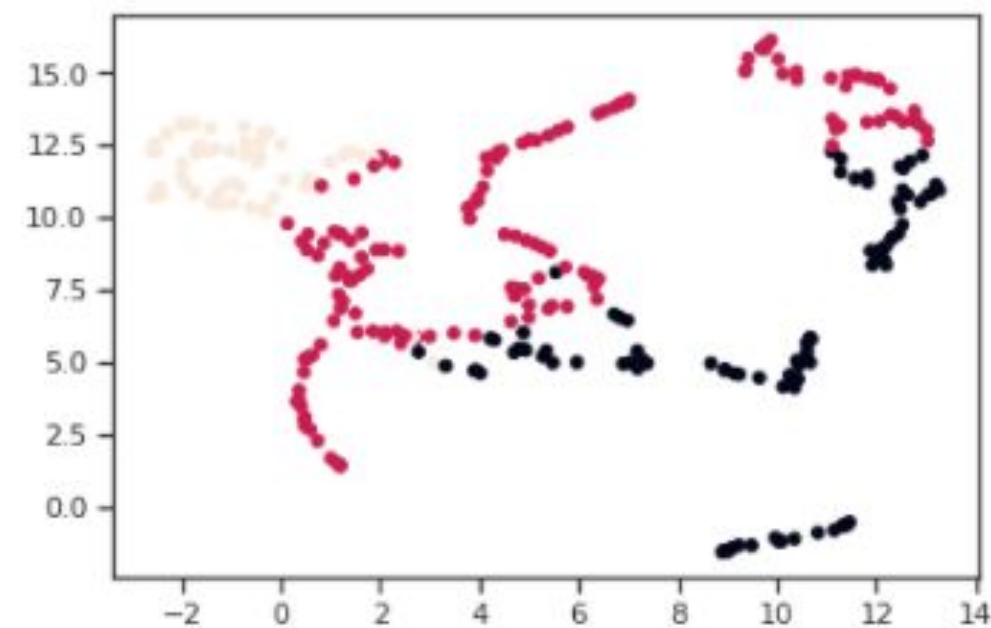
plt.title('Jumlah Customer untuk Setiap Klaster', loc='center', pad=20, fontsize=18, color='black')
sns.countplot(C_kmpp)
C_kmpp[:10]
```

array([0, 2, 2, 1, 1, 1, 2, 1, 2, 1], dtype=int32)

Jumlah Customer untuk Setiap Klaster



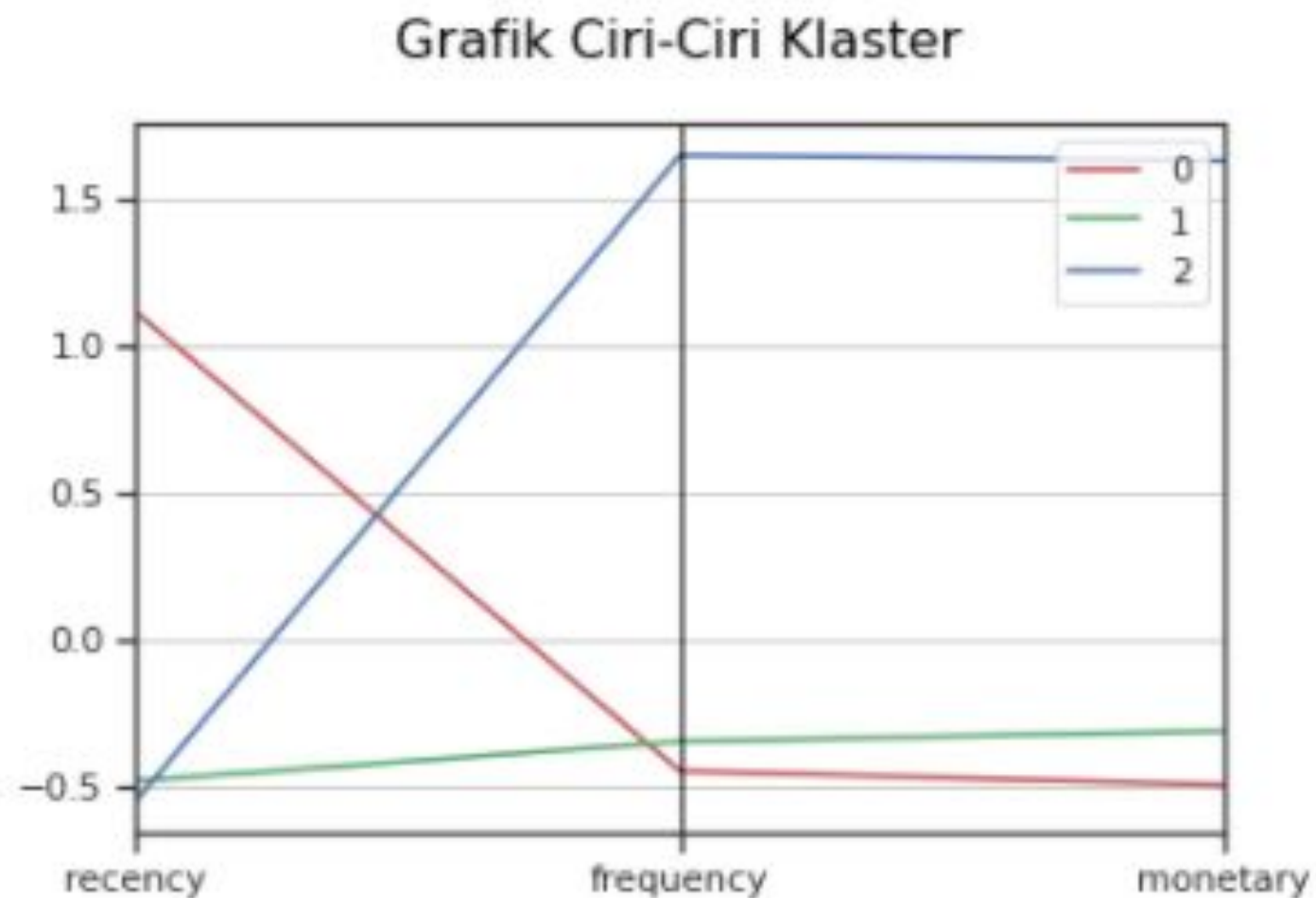
```
#Persebaran klaster
X2D = umap.UMAP(n_neighbors=5, min_dist=0.3, random_state=random_state).fit_transform(scaleddata)
fig, ax = plt.subplots()
ax.scatter(X2D[:,0], X2D[:,1], c=C_kmpp, s=20)
plt.show()
```





```
#Grafik ciri2 klaster
cols = ['recency', 'frequency', 'monetary']
dfC = pd.DataFrame(kmPP.cluster_centers_, columns=cols)
dfC['cluster'] = dfC.index

plt.title('Grafik Ciri-Ciri Klaster', loc='center', pad=20, fontsize=16, color='black')
pd.plotting.parallel_coordinates(dfC, 'cluster', color=('r', 'g', 'b', 'y'))
plt.show()
```

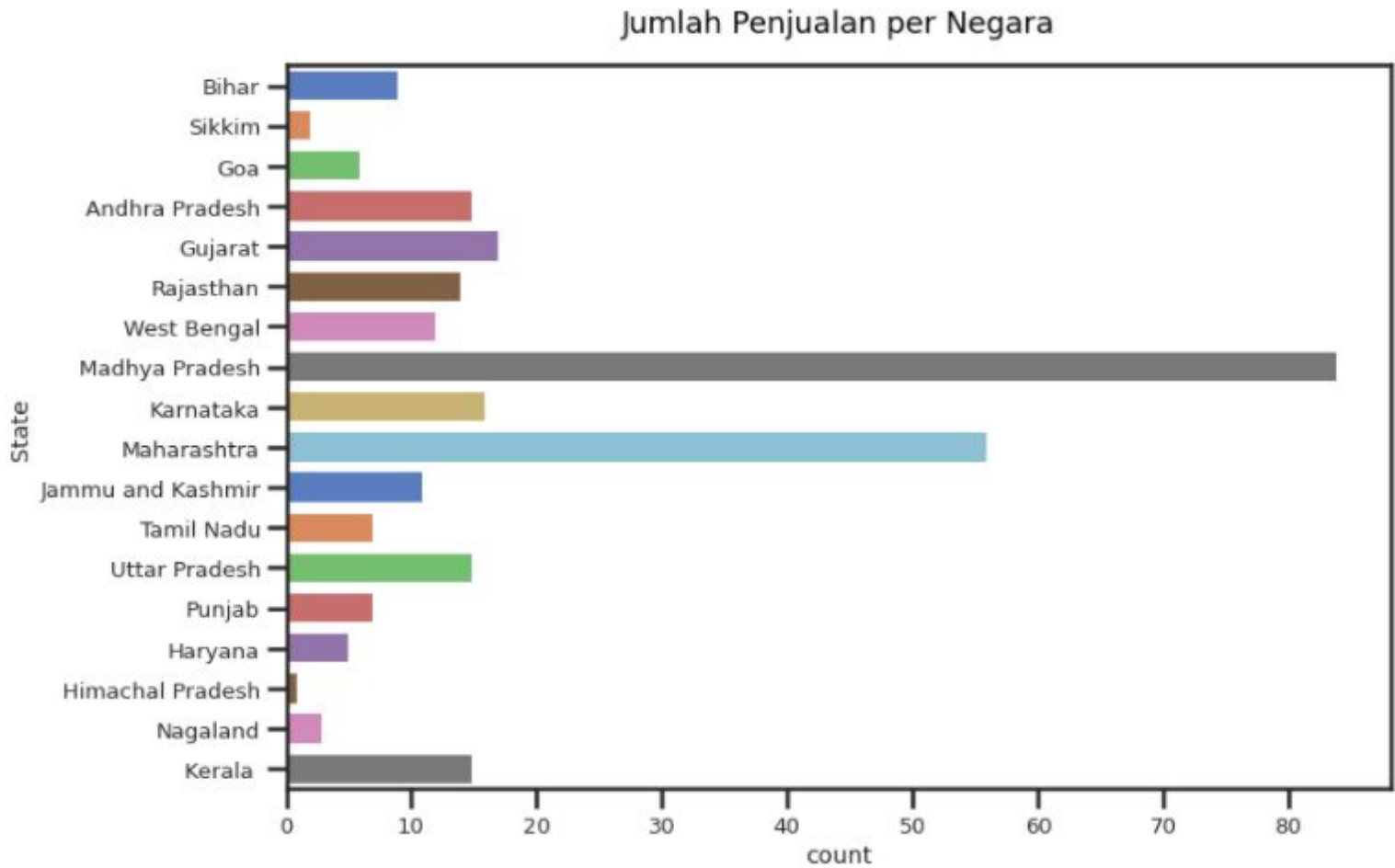
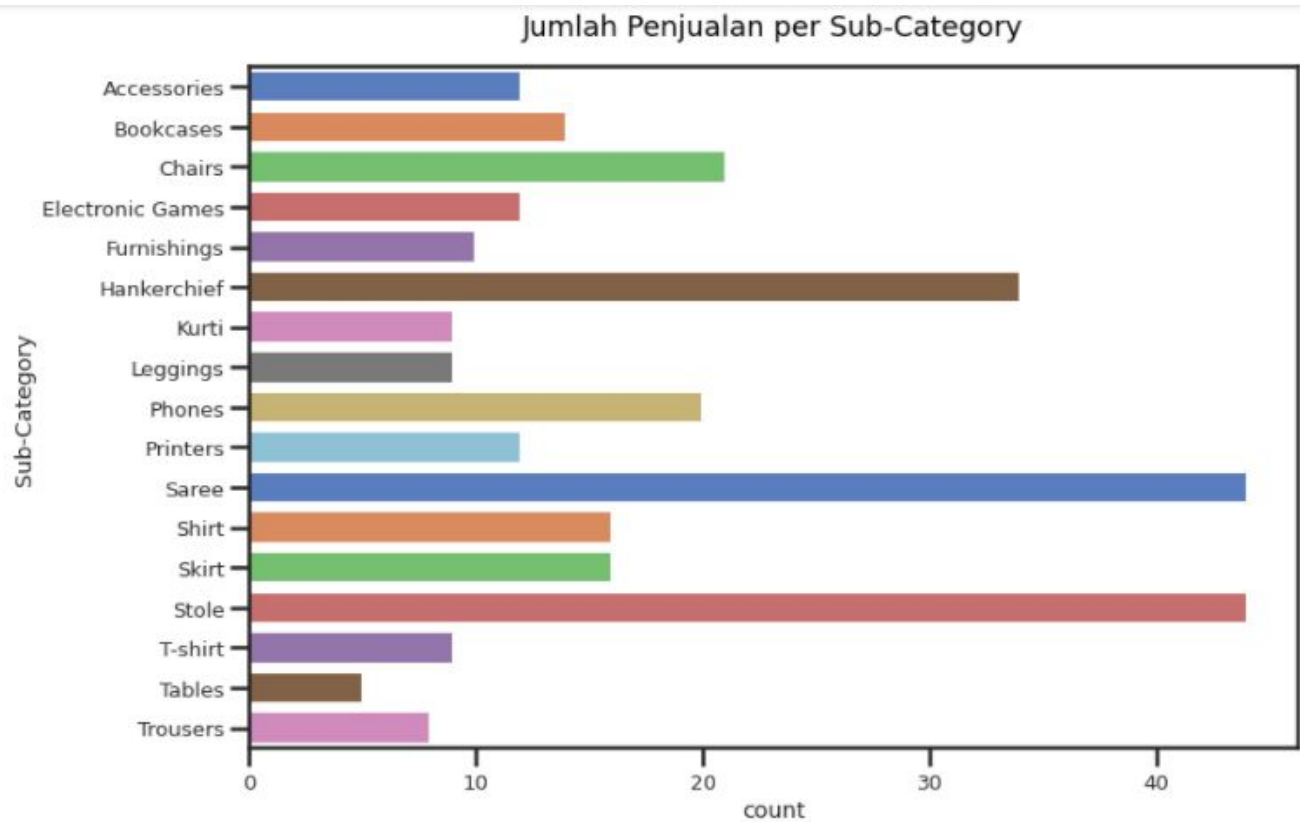
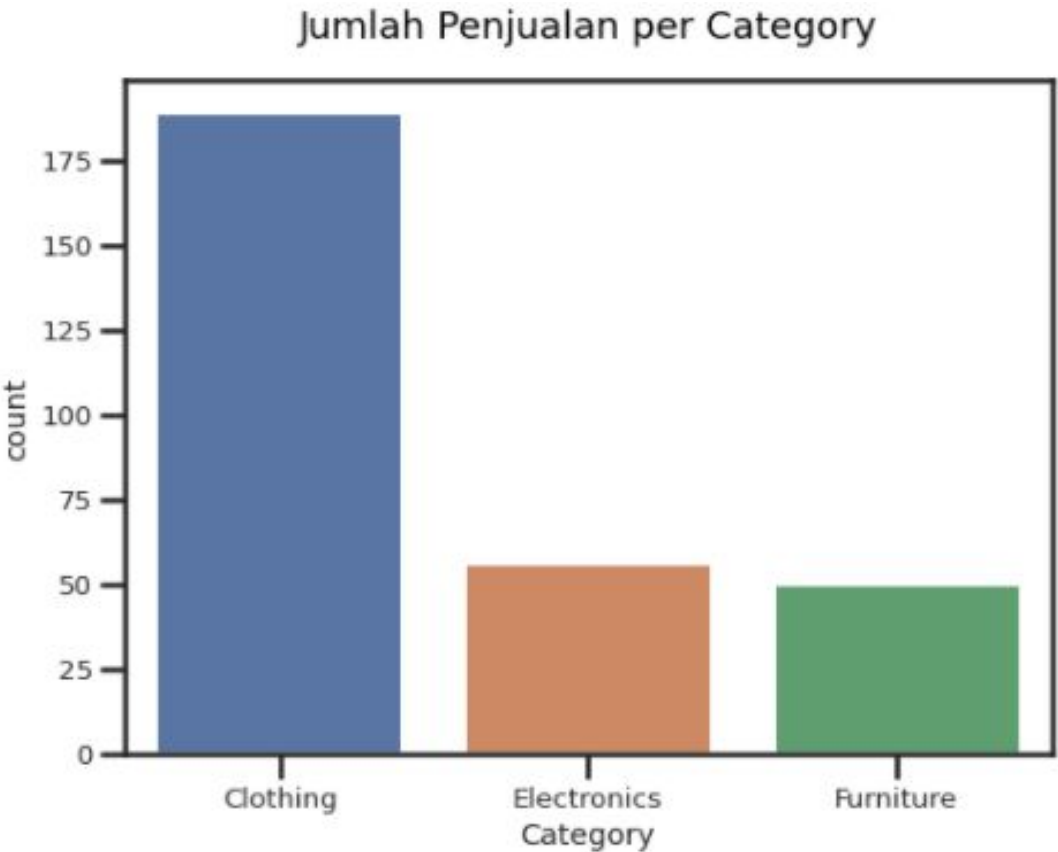


```
#Memisahkan tabel berdasarkan klaster
klaster0 = data2.loc[data2['Cluster']=='0']
klaster1 = data2.loc[data2['Cluster']=='1']
klaster2 = data2.loc[data2['Cluster']=='2']
```

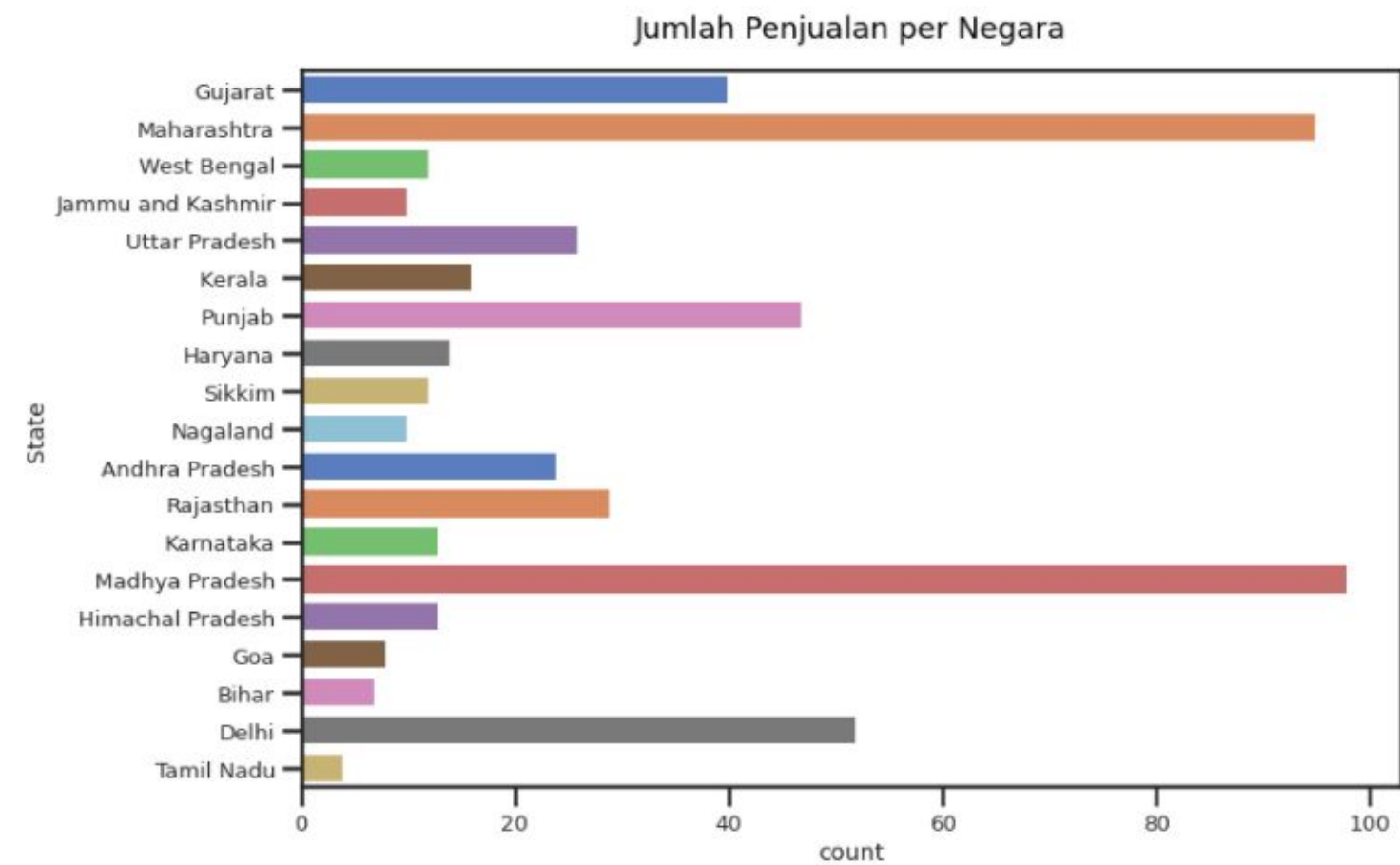
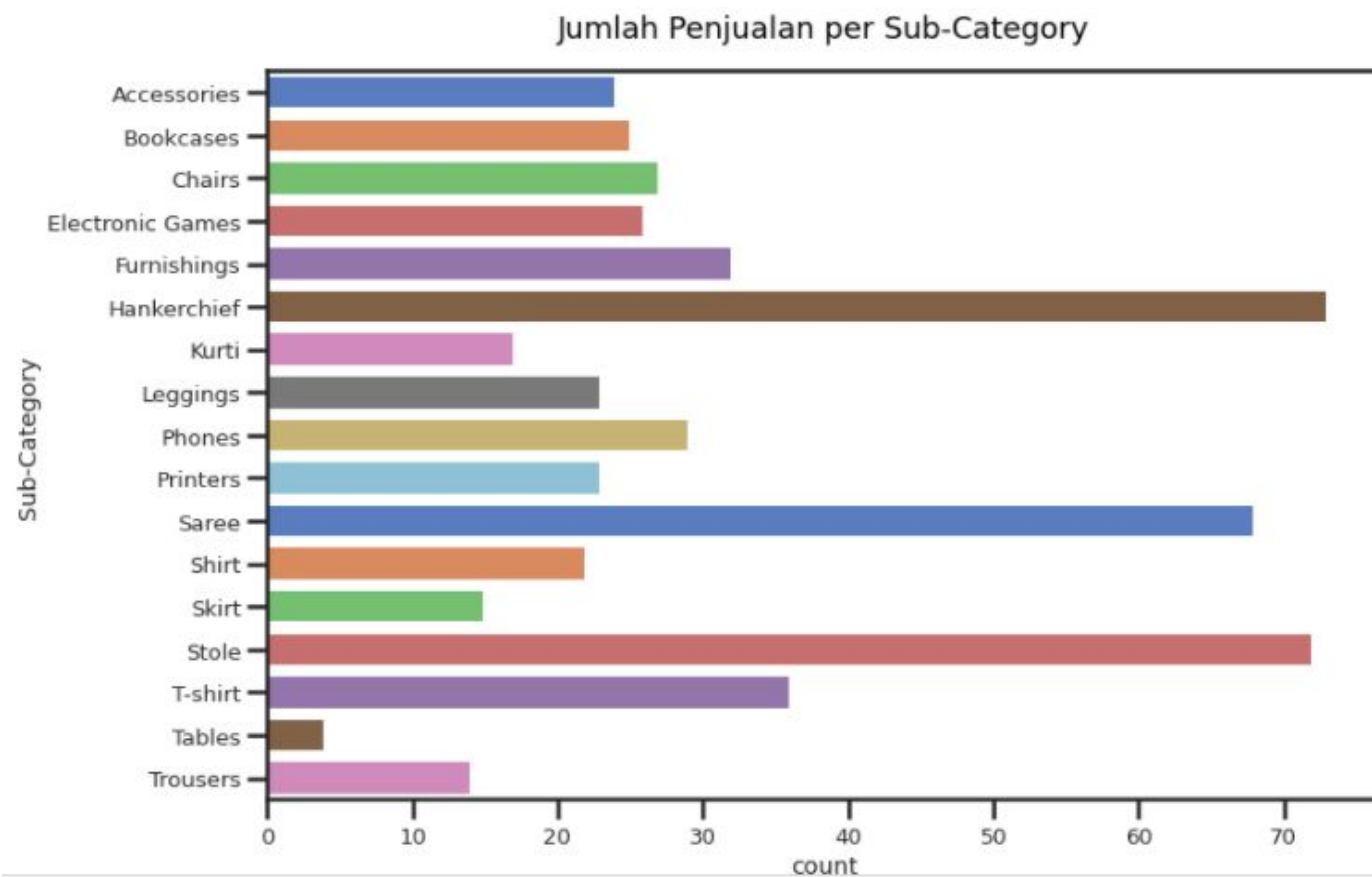
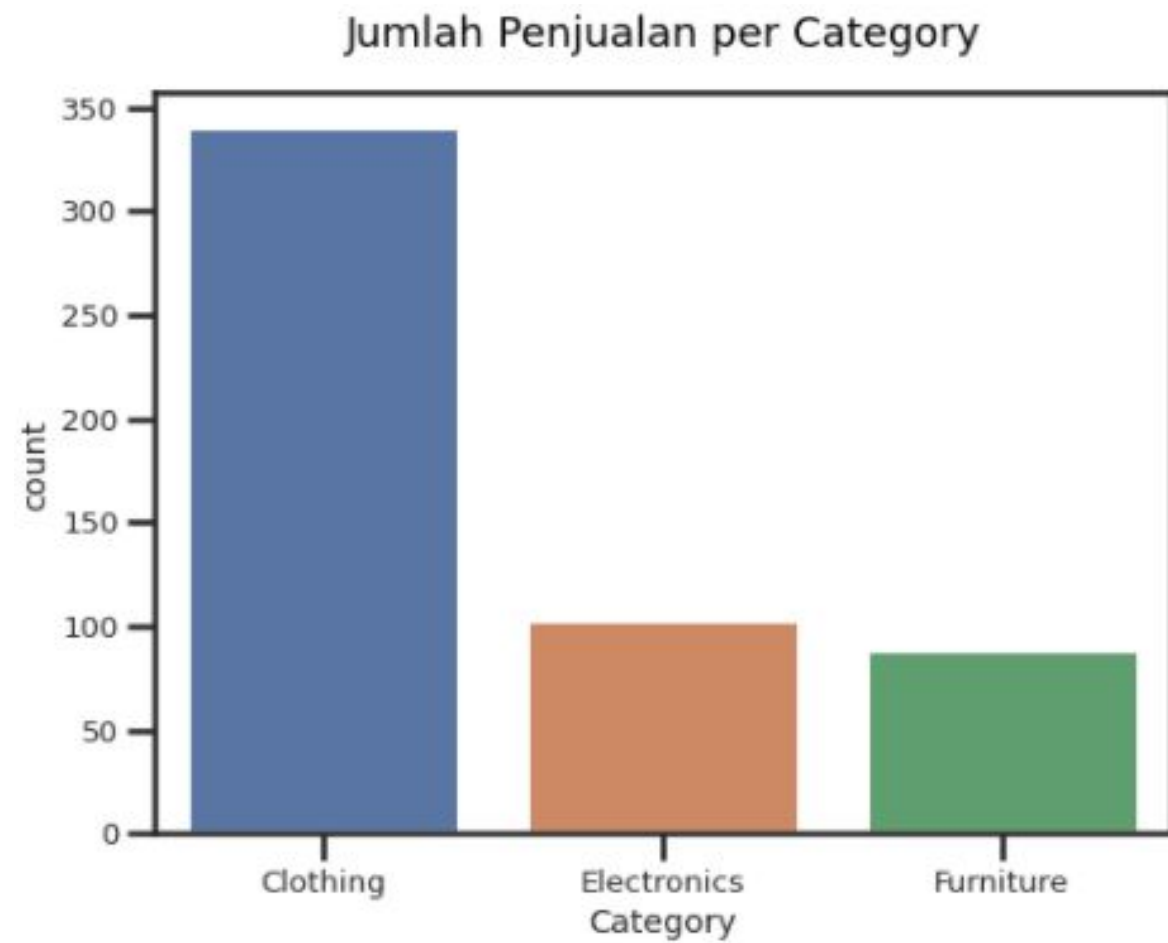
# K-Means++

- **Klaster 0 (Non-Profit):** merupakan customer yang sudah lama membeli produk namun hingga sekarang tidak membeli lagi, frekuensi pembeliannya tidak banyak, begitu pula jumlah nominal uang transaksi juga tidak banyak.
- **Klaster 1 (Silver):** merupakan customer yang baru-baru ini membeli produk, frekuensi pembeliannya jarang tidak begitu banyak namun lebih sering dari klaster 0, jumlah nominal uang transaksi tidak begitu banyak namun lebih baik dari klaster 0 (sedang).
- **Klaster 2 (Gold):** merupakan customer yang baru-baru ini membeli produk, frekuensi pembeliannya banyak, dan jumlah nominal uang transaksinya juga banyak.

# Klaster 0 (Non-Profit)



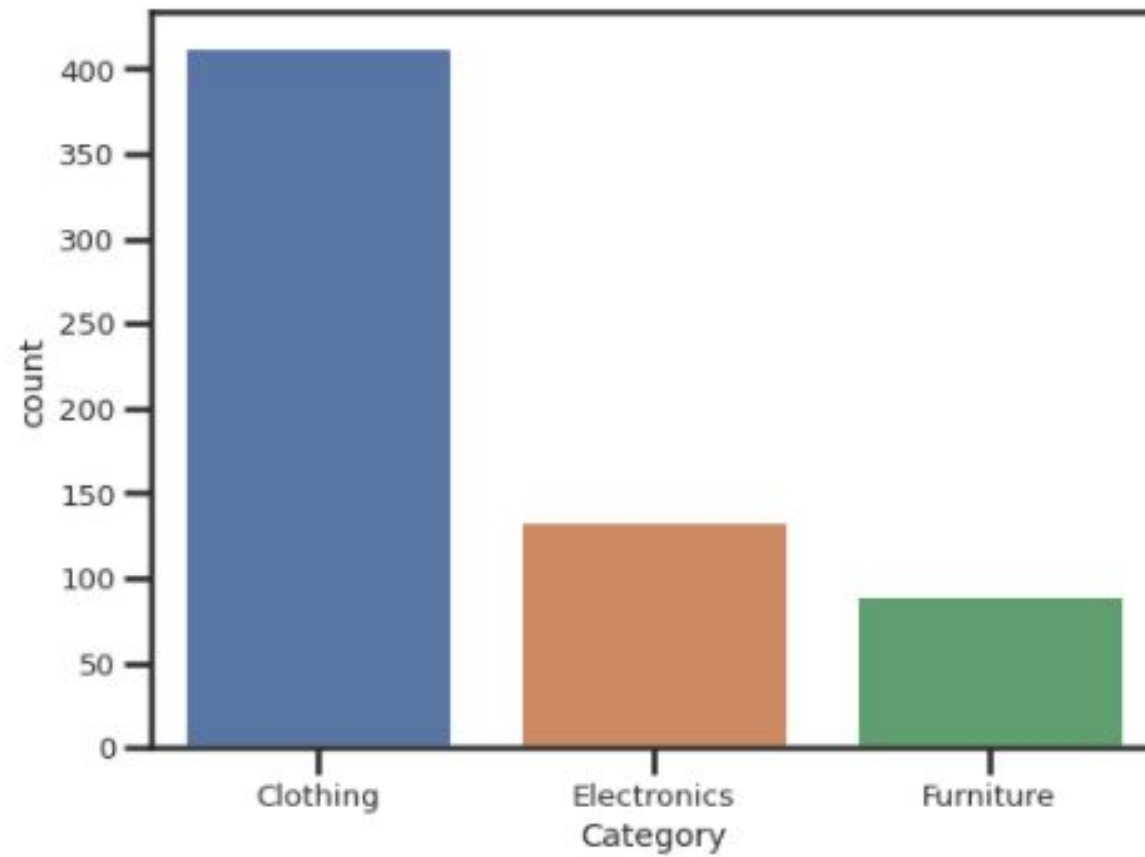
# Klaster 1 (Silver)



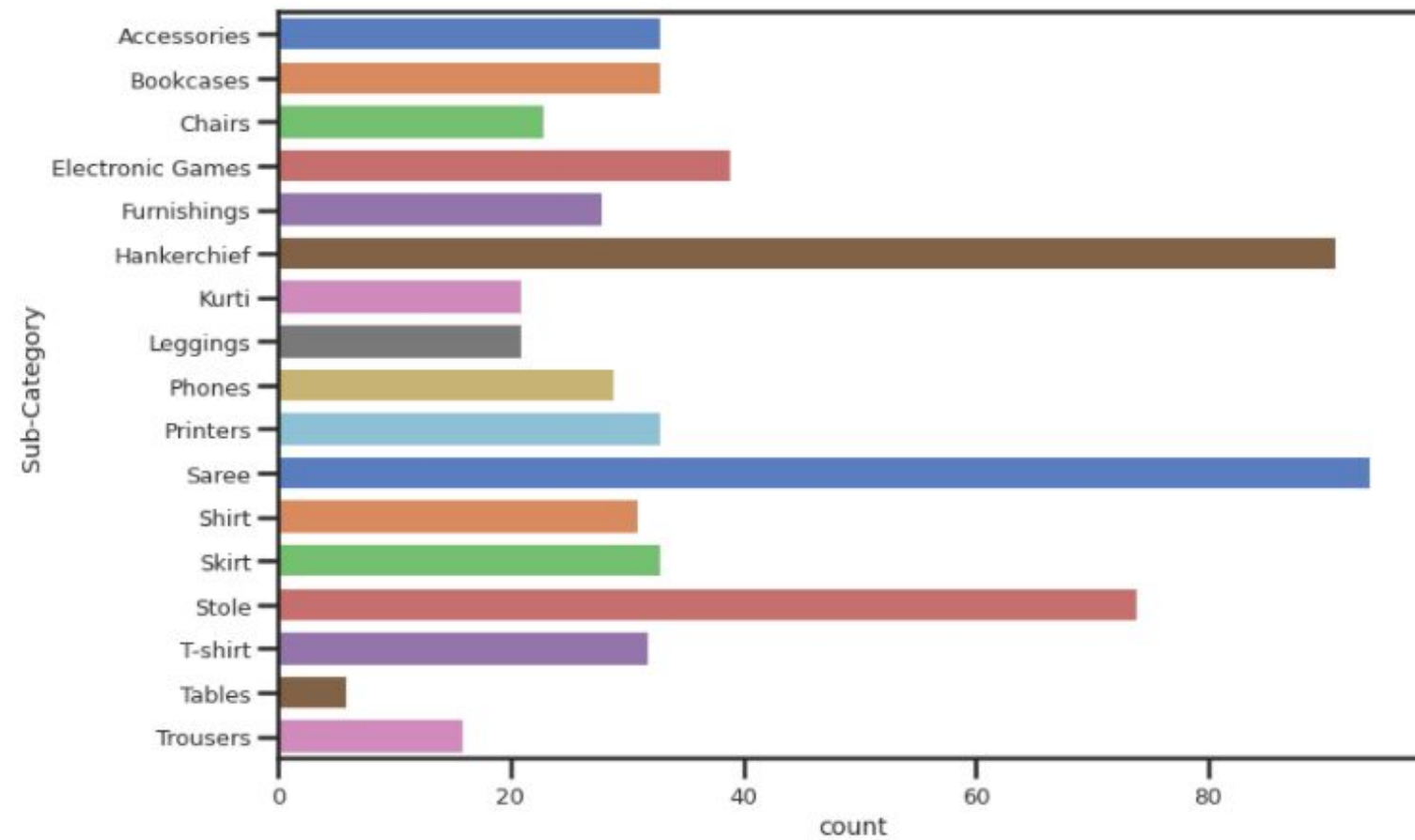


# Klaster 2 (Gold )

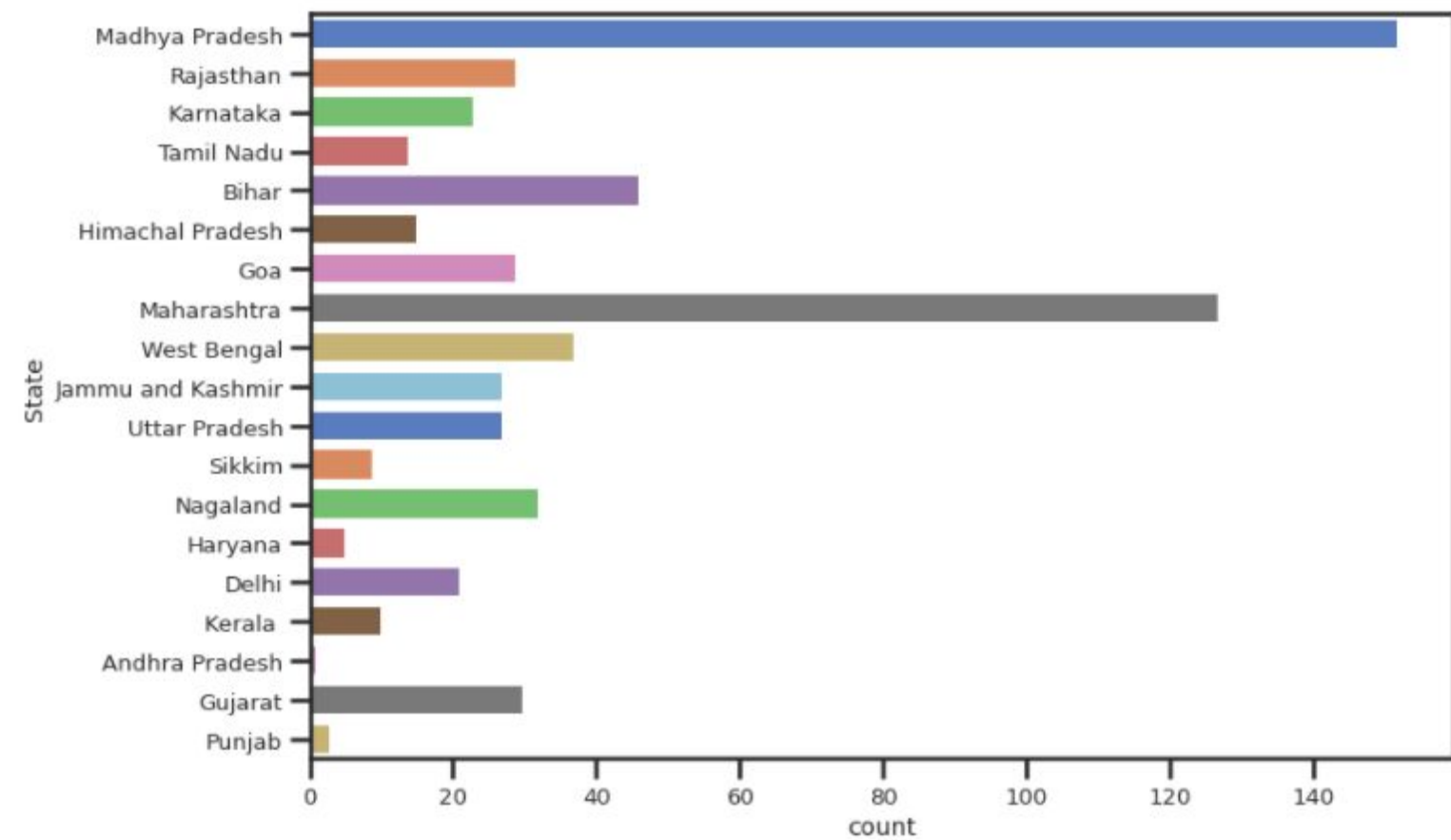
Jumlah Penjualan per Category



Jumlah Penjualan per Sub-Category



Jumlah Penjualan per Negara



# KESIMPULAN

1

Penjualan terbanyak terjadi pada produk pada kategori clothing. Berdasarkan sub-kategori barang yang terjual, penjualan terbanyak ada pada **handkerchief, disusul stole, saree**. Sangat dianjurkan untuk melakukan promo/diskon pada produk tersebut karena penjualannya **keuntungannya yang tinggi**. T-shirt cukup menghasilkan keuntungan yang baik walaupun jumlah penjualannya kurang dari stole. Sehingga T-shirt juga sangat direkomendasikan untuk di promosikan lebih intense lagi.

---

2

Ada **kerugian pada total penjualan table dan chairs** sehingga saya tidak merekomendasikan bagi tim bisnis untuk memberi diskon pada produk-produk tersebut, begitu pula dengan produk **trousers, skirt, legging, kurti, electronic games, phones** untuk saat ini karena keuntungannya masih terlalu sedikit. Kerugian barang pada tiap produk harus dievaluasi dan didiskusikan lagi dengan tim sales, apakah harga pasar turun sehingga mengalami kerugian atau sebab lain.

# KESIMPULAN

3

Keuntungan tertinggi yang didapat berasal dari negara bagian **madhya pradesh, maharashtra, rajasthan, delhi, west bengal, dan gujarat**. Sehingga saya merekomendasikan untuk meningkatkan performa pemasaran pada negara bagian tersebut serta mengevaluasi penjualan pada negara bagian yang lainnya terutama pada negara bagian Haryana dan goa karena keuntungan nya sangat kecil.

---

4

Setelah bulan april 2019, **penjualan mengalami penurunan drastis**, sehingga harus diadakannya evaluasi mengenai pemasaran pada bulan tersebut.



# Rekomendasi

## Gold

**Fokus untuk meningkatkan pembelian customer** sehingga perlu membentuk **cross/Up Selling Strategy**. Cross selling adalah metode untuk menawarkan produk tambahan pada konsumen. Biasanya, yang ditawarkan adalah produk yang sifatnya pelengkap atau terkait dengan produk yang dibeli.

## Silver

**Fokus supaya customer melakukan pembelian kembali** sehingga perlu membentuk **Retention Strategy**. Customer retention adalah upaya untuk **mempertahankan pelanggan yang sudah pernah membeli produk** agar terus melakukan transaksi. Dapat berupa pemberian diskon/promo.

## Non-Profit

Customer pada segmen ini sudah **churn**, maka **fokus campaign untuk mengaktifkan customer kembali dengan membentuk Reactivation strategy**, dengan kata lain strategi pengaktifan kembali. Reactivation strategy ini sendiri bisa **menggunakan reactivation e-mail untuk develop messaging untuk reactivation campaign**.

# Referensi

<https://tau-data.id/>

[dqlab.id](https://dqlab.id)

<https://github.com/joaolcorreia/RFM-analysis/blob/master/RFM%20Analysis.ipynb>

<https://ichi.pro/id/histogram-dan-plot-massa-jenis-dengan-python-271294437219008>

<https://gifadn.medium.com/data-manipulation-with-pandas-part-i-10ba57dd9e06>

<https://stackoverflow.com/questions/23199796/detect-and-exclude-outliers-in-pandas-data-frame>

<https://stackoverflow.com/questions/51603485/rfm-analysis-by-month-in-pandas>

<https://answer-id.com/id/51373975>

<https://stackoverflow.com/questions/60624571/sort-list-of-month-name-strings-in-ascending-order>

<https://www.kaggle.com/aashirgurung/sales-analysis>

<http://e-journal.uajy.ac.id/11383/3/2EM19690.pdf>

[https://www.researchgate.net/publication/326849650\\_PENERAPAN\\_METODE\\_K-MEANS\\_CLUSTERING\\_PADA\\_PERUSAHAAN](https://www.researchgate.net/publication/326849650_PENERAPAN_METODE_K-MEANS_CLUSTERING_PADA_PERUSAHAAN)

<http://shirleymelita.blog.uma.ac.id/wp-content/uploads/sites/151/2018/11/ANALISIS-KORELAS.pdf>

<https://stackoverflow.com/questions/4700614/how-to-put-the-legend-out-of-the-plot>

<https://rpubs.com/gabrielerichson/custmarkets-p2>

<https://www.niagahoster.co.id/blog/cross-selling-dan-up-selling/>

<https://glints.com/id/lowongan/strategi-customer-retention/#.YVf3-ppBzIU>

<https://www.omnisend.com/blog/customer-reactivation-marketing-why-and-how-to-do-it/>

PENGARUH CAR, FDR DAN OER TERHADAP PROFITABILITAS (ROA) PADA BANK UMUM SYARIAH DI INDONESIA PERIODE 2011-2015 Erika Bellani ,  
Kardinal - Jurusan Manajemen STIE Multi Data Palembang

Terima Kasih