Super Market dataset

ALP _ data mining

Bernadus Sergio Halim - 0706022210056

Gio Vanni Elbert Wisman - 0706022210034

About dataset

supermarket

Dataset Superstore ini mencakup data transaksi, pelanggan, dan pengembalian barang yang terbagi dalam tiga sheet: Orders, People, dan Returns. Dataset ini memberikan gambaran menyeluruh tentang dinamika ritel, mulai dari analisis penjualan, profitabilitas, dan segmentasi pelanggan hingga tren pengembalian barang. Dengan data ini, kita dapat mengeksplorasi wawasan bisnis seperti prediksi penjualan, analisis produk terlaris, hingga pengembangan strategi untuk meningkatkan retensi pelanggan dan mengurangi pengembalian. Cocok untuk analisis mendalam dan aplikasi praktis dalam pengambilan keputusan bisnis. Dataset ini terdapat 21 column dan lebih dari 1000 baris

isi column dataset

Row ID Deskripsi: Nomor baris unik untuk setiap catatan dalam dataset. Tipe Data: Integer

Order ID Deskripsi: Identifikasi unik untuk setiap pesanan. Tipe Data: String

Order Date Deskripsi: Tanggal ketika pesanan dibuat. Tipe Data: Date Format: YYYY-MM-DD

Ship Date Deskripsi: Tanggal pengiriman pesanan. Tipe Data: Date Format: YYYY-MM-DD

Ship Mode Deskripsi: Metode pengiriman pesanan. Tipe Data: String

Customer ID Deskripsi: Identifikasi unik pelanggan. Tipe Data: String

Customer Name Deskripsi: Nama pelanggan yang melakukan pesanan. Tipe Data: String

Segment Deskripsi: Segmen pasar tempat pelanggan berada. Tipe Data: String

Country/Region Deskripsi: Negara atau wilayah tempat pesanan dilakukan. Tipe Data: String

City Deskripsi: Kota tujuan pengiriman pesanan. Tipe Data: String

State/Province Deskripsi: Negara bagian atau provinsi tujuan pengiriman. Tipe Data: String

Postal Code Deskripsi: Kode pos tujuan pengiriman. Tipe Data: Integer/String

Region Deskripsi: Wilayah geografis tempat pesanan diklasifikasikan. Tipe Data: String

Product ID Deskripsi: Identifikasi unik produk yang dipesan. Tipe Data: String

Category Deskripsi: Kategori produk. Tipe Data: String

Sub-Category Deskripsi: Subkategori produk. Tipe Data: String

Product Name Deskripsi: Nama produk yang dipesan. Tipe Data: String

Sales Deskripsi: Total penjualan dari produk tertentu. Tipe Data: Float

Quantity Deskripsi: Jumlah unit produk yang dipesan. Tipe Data: Integer

Discount Deskripsi: Diskon yang diterapkan pada pesanan. Tipe Data: Float

Profit Deskripsi: Keuntungan yang diperoleh dari pesanan. Tipe Data: Float

Data Cleaning and Preperation

In [12]:
```
# Pastikan tidak ada nilai yang hilang setelah transformasi
print("\nJumlah Missing Values Setelah Transformasi:")
data_cleaned.isnull().sum()
```

```
Jumlah Missing Values Setelah Transformasi:
```

Out[12]:

Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
Country/Region	0
City	0
State/Province	0
Postal Code	0
Region	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0
Sales	0
Quantity	0
Discount	0
Profit	0

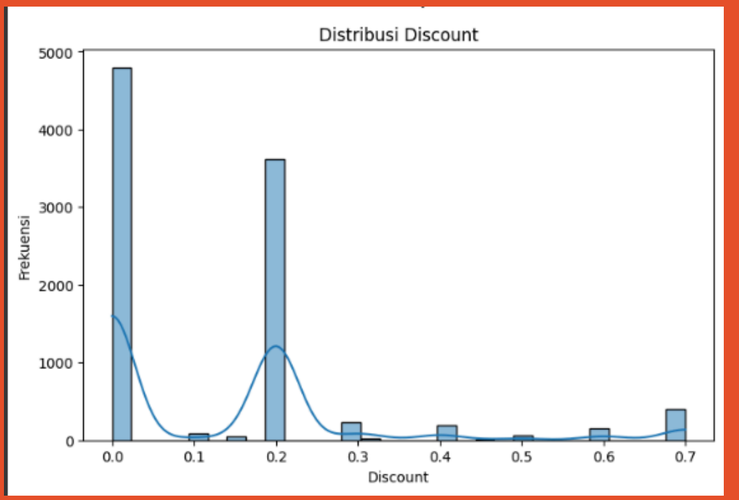
dtype: int64

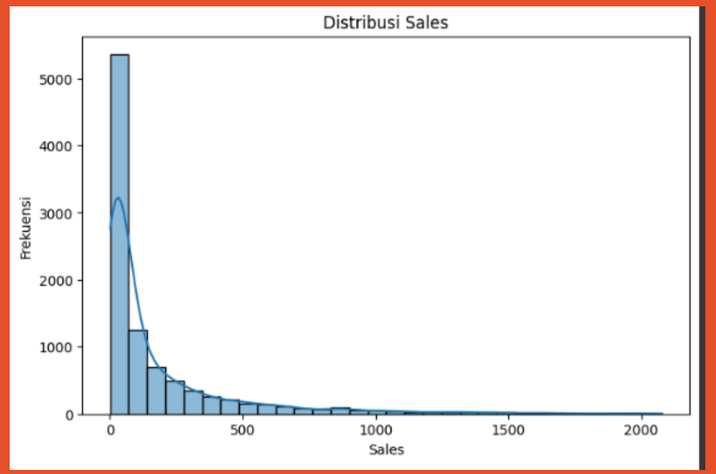
In [13]:
```
# Dataset setelah preparation
print("\nDataset Setelah Data Preparation:")
data_cleaned.info()
```

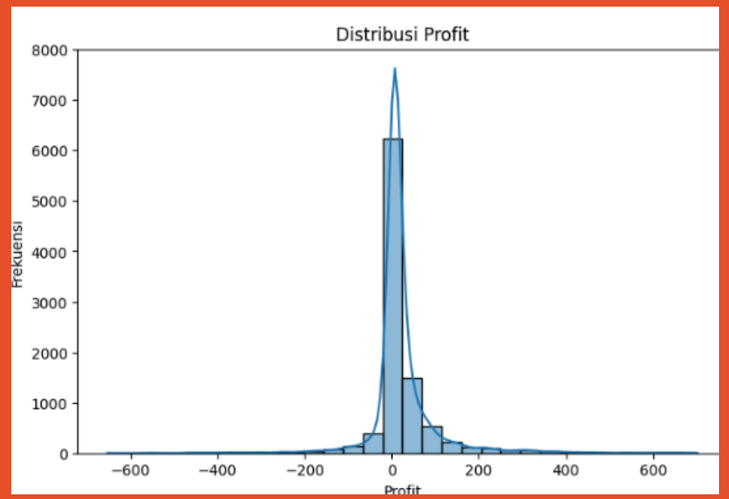
```
Dataset Setelah Data Preparation:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9633 entries, 0 to 10193
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          9633 non-null   int64
 1   Order ID        9633 non-null   object
 2   Order Date      9633 non-null   datetime64[ns]
 3   Ship Date       9633 non-null   datetime64[ns]
 4   Ship Mode       9633 non-null   int64
 5   Customer ID     9633 non-null   object
 6   Customer Name   9633 non-null   object
 7   Segment         9633 non-null   category
 8   Country/Region  9633 non-null   object
 9   City            9633 non-null   object
 10  State/Province  9633 non-null   object
 11  Postal Code     9633 non-null   object
 12  Region          9633 non-null   object
 13  Product ID      9633 non-null   object
 14  Category        9633 non-null   object
 15  Sub-Category    9633 non-null   object
 16  Product Name    9633 non-null   object
 17  Sales           9633 non-null   float64
 18  Quantity        9633 non-null   int64
 19  Discount        9633 non-null   float64
 20  Profit          9633 non-null   float64
dtypes: category(1), datetime64[ns](2), float64(3), int64(3), object(12)
memory usage: 1.6+ MB
```

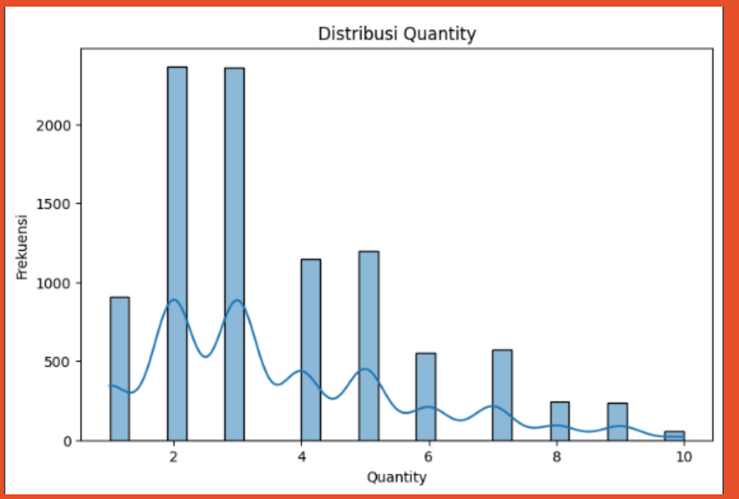
Histogram for numeric data distribution

In [15]:
```
# Columns for analysis
numeric_columns = ['Sales', 'Quantity', 'Discount', 'Profit']
# Histogram for numeric data distribution
for col in numeric_columns:
    plt.figure(figsize=(8, 5))
    sns.histplot(data_cleaned[col], kde=True, bins=30)
    plt.title(f'Distribusi {col}')
    plt.xlabel(col)
    plt.ylabel('Frekuensi')
    plt.show()
```



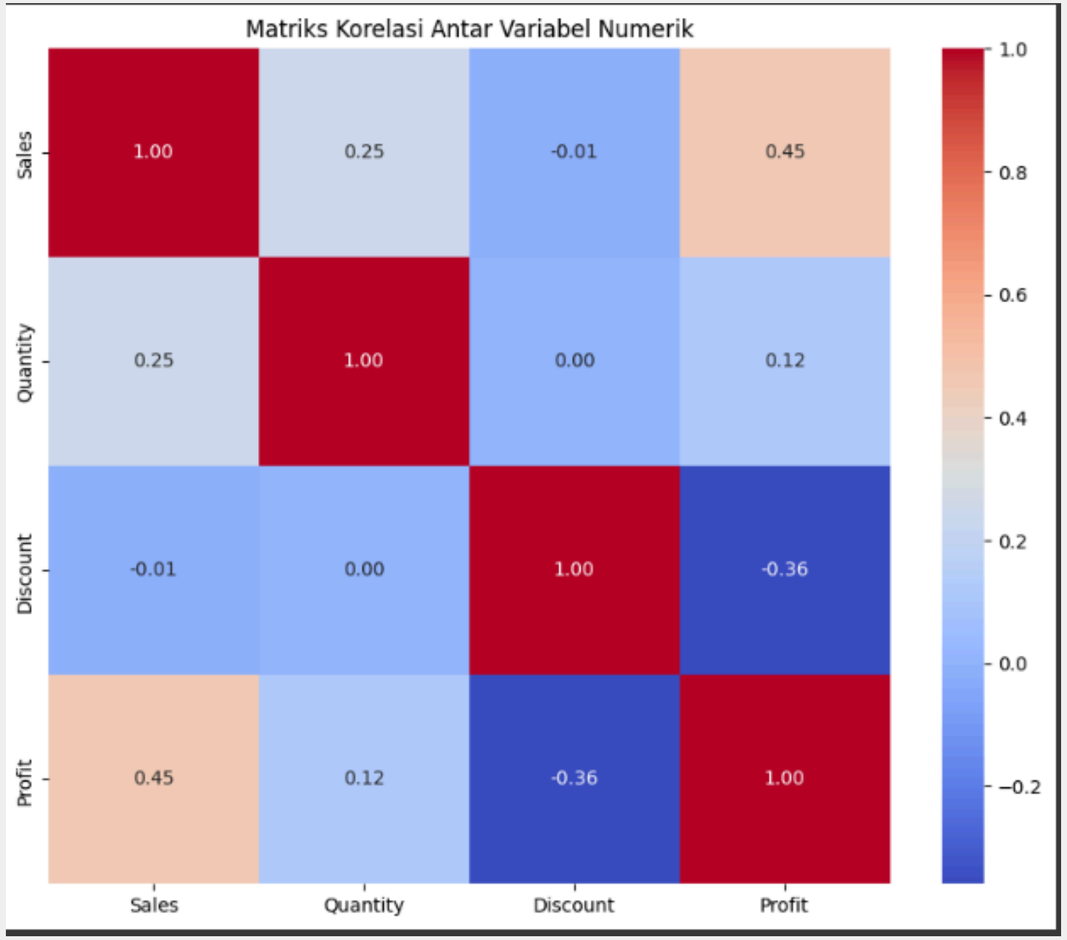






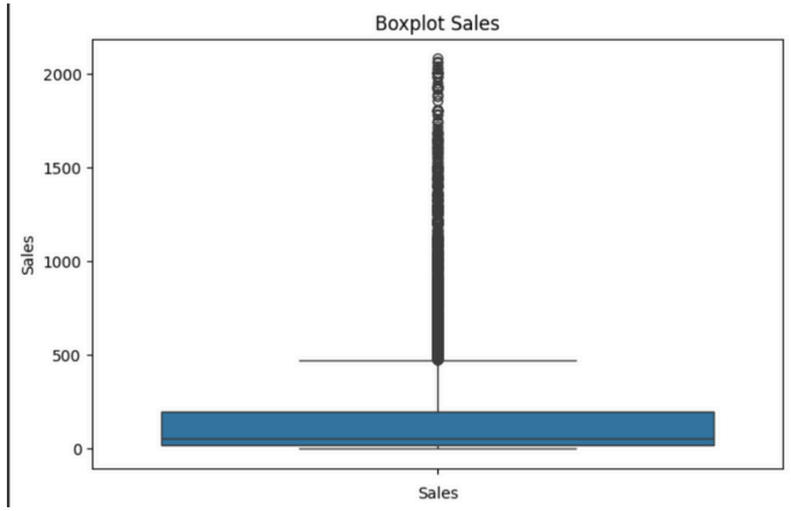
Correlation matrix for numeric variables

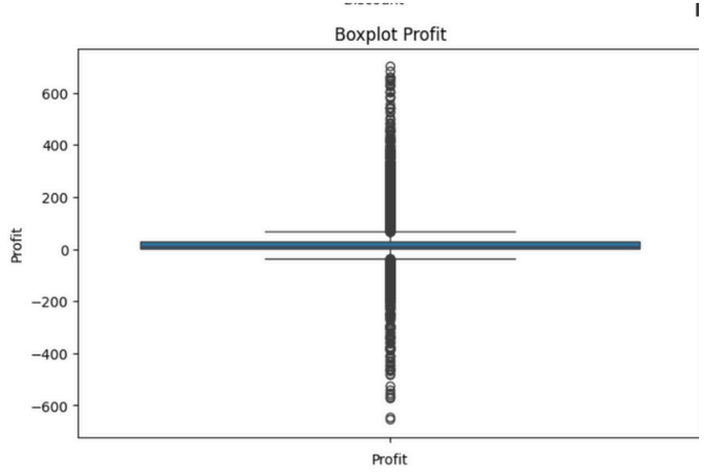
```
# Correlation matrix for numeric variables
plt.figure(figsize=(10, 8))
correlation_matrix = data_cleaned[numeric_columns].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Matriks Korelasi Antar Variabel Numerik')
plt.show()
```

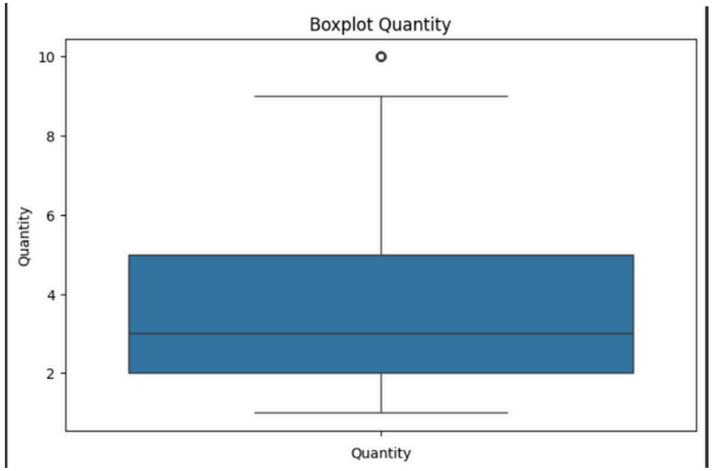


Boxplot for outliers in numeric data

In [16]:
```
# Boxplot for outliers in numeric data
for col in numeric_columns:
    plt.figure(figsize=(8, 5))
    sns.boxplot(data=data_cleaned[col])
    plt.title(f'Boxplot {col}')
    plt.xlabel(col)
    plt.show()
```

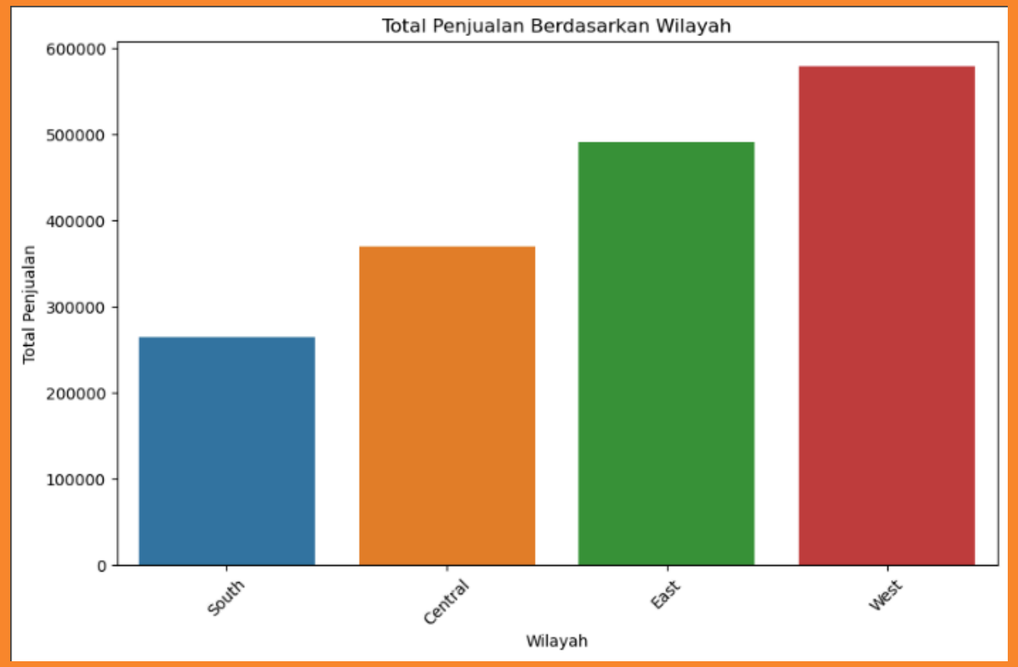






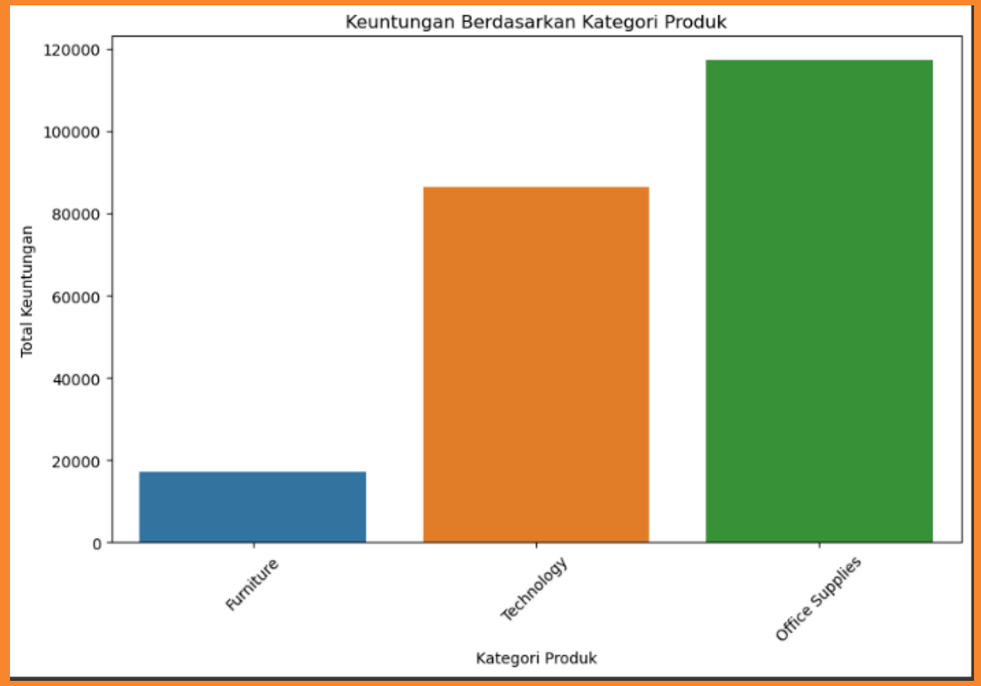
Total penjualan berdasarkan wilayah

```
# Sales trends by region
plt.figure(figsize=(10, 6))
region_sales = data_cleaned.groupby('Region')['Sales'].sum().sort_values()
sns.barplot(x=region_sales.index, y=region_sales.values)
plt.title('Total Penjualan Berdasarkan Wilayah')
plt.xlabel('Wilayah')
plt.ylabel('Total Penjualan')
plt.xticks(rotation=45)
plt.show()
```



keuntungan berdasarkan kategori produk

```python
# Profit trends by product category
plt.figure(figsize=(10, 6))
category_profit = data_cleaned.groupby('Category')['Profit'].sum().sort_values()
sns.barplot(x=category_profit.index, y=category_profit.values)
plt.title('Keuntungan Berdasarkan Kategori Produk')
plt.xlabel('Kategori Produk')
plt.ylabel('Total Keuntungan')
plt.xticks(rotation=45)
plt.show()
```



Feature Engineering

```python
from sklearn.preprocessing import StandardScaler, MinMaxScaler, LabelEncoder

# 4. Feature Engineering

# Encoding kolom kategori
categorical_columns = ['Region', 'Category', 'Sub-Category', 'Segment']
label_encoders = {}
for col in categorical_columns:
    le = LabelEncoder()
    data_cleaned[col] = le.fit_transform(data_cleaned[col])
    label_encoders[col] = le  # Simpan encoder untuk referensi jika diperlukan

# Standarisasi data numerik
scaler = StandardScaler()
numeric_columns = ['Sales', 'Quantity', 'Discount', 'Profit']
data_cleaned[numeric_columns] = scaler.fit_transform(data_cleaned[numeric_columns])

# Normalisasi data numerik (opsional, jika diperlukan sebagai alternatif standarisasi)
# scaler_minmax = MinMaxScaler()
# data_cleaned[numeric_columns] = scaler_minmax.fit_transform(data_cleaned[numeric_columns])

# Menampilkan dataset setelah feature engineering
print("\nDataset Setelah Feature Engineering:")
print(data_cleaned.head())

# Simpan dataset yang sudah diolah (opsional)
data_cleaned.to_csv("processed_superstore.csv", index=False)
```

```
Dataset Setelah Feature Engineering:
   Row ID        Order ID Order Date  Ship Date  Ship Mode Customer ID  \
0       1  US-2021-103800 2021-03-01 2021-07-01        NaN    DP-13000
2       3  US-2021-112326 2021-04-01 2021-08-01        NaN    PO-19195
3       4  US-2021-112326 2021-04-01 2021-08-01        NaN    PO-19195
4       5  US-2021-141817 2021-05-01 2021-12-01        NaN    MB-18085
6       7  US-2021-167199 2021-06-01 2021-10-01        NaN    ME-17320

   Customer Name  Segment Country/Region         City  ... Postal Code  \
0  Darren Powers        0  United States      Houston  ...       77095
2  Phillina Ober        2  United States   Naperville  ...       60540
3  Phillina Ober        2  United States   Naperville  ...       60540
4     Mick Brown        0  United States  Philadelphia  ...       19143
6  Maria Etezadi        2  United States    Henderson  ...       42420

  Region       Product ID Category  Sub-Category  \
0      0  OFF-PA-10000174        1            12
2      0  OFF-LA-10003223        1            10
3      0  OFF-ST-10002743        1            14
4      1  OFF-AR-10003478        1             2
6      2  OFF-AR-10001662        1             2

                                        Product Name     Sales  Quantity  \
0  Message Book, Wirebound, Four 5 1/2" X 4" Form... -0.550584 -0.821827
2                                      Avery 508 -0.566590 -0.325702
3                  SAFCO Boltless Steel Shelving  0.328987 -0.325702
4  Avery Hi-Liter EverBold Pen Style Fluorescent ... -0.539986 -0.325702
6           Rogers Handheld Barrel Pencil Sharpener -0.588226 -0.821827

   Discount    Profit
0  0.369059 -0.216227
2  0.369059 -0.232161
3  0.369059 -1.091986
4  0.369059 -0.224536
6 -0.774387 -0.266930

[5 rows x 21 columns]
C:\Users\GEOVANI\AppData\Local\Temp\ipykernel_24236\1940379765.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

Evaluasi Model Klasifikasi dan Clustering

Hasil Evaluasi Model Klasifikasi:

	precision	recall	f1-score	support
0	0.54	0.66	0.59	1010
1	0.36	0.29	0.32	591
2	0.18	0.12	0.15	326
accuracy			0.45	1927
macro avg	0.36	0.36	0.35	1927
weighted avg	0.42	0.45	0.43	1927

Centroid Clusters:

```
[[ 0.2044882   0.11046747  2.30226715 -1.16833036]
 [ 2.53134264  0.71129511 -0.35011646  2.16930913]
 [-0.27168684 -0.08302666 -0.25629361 -0.06341199]]
```

Jumlah anggota setiap cluster:

```
[ 993  765 7875]
```

```python
X_cluster = data_cleaned[['Sales', 'Quantity', 'Profit']]


# Choose k based on above
optimal_k = 3
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
data_cleaned['Cluster'] = kmeans.fit_predict(X_cluster)

# 6. Evaluasi Model

# Evaluasi Model 1: Klasifikasi
accuracy = accuracy_score(y_test, y_pred)
print("Evaluasi Model Klasifikasi:")
print(f"Accuracy: {accuracy:.2f}")

# Evaluasi Model 2: Clustering
silhouette_avg = silhouette_score(X_cluster, data_cleaned['Cluster']) # use the selected features
print("\nEvaluasi Model Clustering:")
print(f"Silhouette Score: {silhouette_avg:.2f}")

# Bandingkan hasil kedua model
print("\nPerbandingan Model:")
print(f"Model Klasifikasi: Accuracy = {accuracy:.2f}")
print(f"Model Clustering: Silhouette Score = {silhouette_avg:.2f}")
```

```
<ipython-input-36-72d8dae7a479>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexi
  data_cleaned['Cluster'] = kmeans.fit_predict(X_cluster)
Evaluasi Model Klasifikasi:
Accuracy: 0.52

Evaluasi Model Clustering:
Silhouette Score: 0.47

Perbandingan Model:
Model Klasifikasi: Accuracy = 0.52
Model Clustering: Silhouette Score = 0.47
```

Thank You