TUGAS PRESENTASI DATA MINING

Klasifikasi Data Penjualan Menggunakan Metode Market Basket Analysis

Kelompok 1 :

Iqbal Octoryaz - 22171065003

Semayatri F. Tunliu - 22271065704

Riko Adytia . A - 23171065708

DESKRIPSI DATASET

Analisis Market Base dengan algoritma Apriori

Distributor ingin menargetkan pelanggan dengan saran mengenai kumpulan item yang kemungkinan besar akan dibeli oleh pelanggan. Lalu kumpulan data dari Distributor yang berisi data transaksi memberikan data seputar semua transaksi yang telah terjadi dalam periode waktu tertentu. Kemudian distributor akan menggunakan hasil untuk mengembangkan industrinya dan memberikan saran kepada pelanggan mengenai item yang dapat dibeli, Keterlibatan pelanggan dalam meningkatkan pengalaman berbelanja. Dengan menggunakan aturan asosiasi yang memeriksa ketergantungan/hubungan satu item data pada item data lainnya.

DESKRIPSI DATASET

Analisis Market Base dengan algoritma Apriori

Kami ingin menargetkan pelanggan dengan saran mengenai kumpulan item yang kemungkinan besar akan dibeli oleh pelanggan. Kami diberi kumpulan data yang berisi :

Nama Dataset : Assignment-1_Data.xlsx.
Sumber Data : Kaggle.
Tujuan Dataset : Dataset ini digunakan untuk menganalisis perilaku pembelian dan seberapa sering item tersebut terjual.

Kami akan menggunakan dataset yang ada untuk memberikan saran pelanggan mengenai item, kami dapat meningkatkan keterlibatan pelanggan dalam meningkatkan pengalaman pemesanan barang serta mengidentifikasi perilaku pelanggan.

Deskripsi Variabel

Pada Dataset Assignment data terdapat 7 kolom dengan 4 Tipe Data sbb:

No	Nama	Tipe Data	Deskripsi
1	BillNo	Object	Nomor Bill Pada Setiap Transaksi
2	Itemname	Object	Nama Item pemesanan pada bill
3	Quantity	Integer	Jumlah barang yang dipesan setiap item
4	Date	Date Time	Tanggal Pemesanan/transaksi barang
5	Price	Float	Harga dari setiap Item pemesanan
6	CustomerID	Float	Nomor id dari pelanggan yang memesan barang
7	Country	Object	Negara yang melakukan pemesanan/transaksi barang

Data Preprocessing

Mengimport library yang dibutuhkan

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import plotly.express as px
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

import warnings
warnings.filterwarnings('ignore')
```

Mengimport dataset

1. Loading data

```
[3]
df=pd.read_excel(r'C:\Users\Iqbal\Downloads\archive (10)\Assignment-1_Data.xlsx')
df.head()
```

...

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Karakteristik data

```
df.describe()
```

	Quantity	Price	CustomerID
count	522064.000000	522064.000000	388023.000000
mean	10.090435	3.826801	15316.931710
std	161.110525	41.900599	1721.846964
min	-9600.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13950.000000
50%	3.000000	2.080000	15265.000000
75%	10.000000	4.130000	16837.000000
max	80995.000000	13541.330000	18287.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 522064 entries, 0 to 522063
Data columns (total 7 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   BillNo      522064 non-null  object
 1   Itemname    520609 non-null  object
 2   Quantity    522064 non-null  int64
 3   Date        522064 non-null  datetime64[ns]
 4   Price       522064 non-null  float64
 5   CustomerID  388023 non-null  float64
 6   Country     522064 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 27.9+ MB
```

Menghapus kolom Quantity yang memiliki nilai <= 0

```
df[(df.Quantity<=0)].head()
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
2359	536589	NaN	-10	2010-12-01 16:50:00	0.0	NaN	United Kingdom
4289	536764	NaN	-38	2010-12-02 14:42:00	0.0	NaN	United Kingdom
6998	536996	NaN	-20	2010-12-03 15:30:00	0.0	NaN	United Kingdom
6999	536997	NaN	-20	2010-12-03 15:30:00	0.0	NaN	United Kingdom
7000	536998	NaN	-6	2010-12-03 15:30:00	0.0	NaN	United Kingdom

```
df = df[~(df.Quantity<0)]
df[(df.Quantity<=0)].head()
```

BillNo	Itemname	Quantity	Date	Price	CustomerID	Country

Menghapus kolom Price yang memiliki nilai <= 0

```
df[(df.Price==0)].head()
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
613	536414	NaN	56	2010-12-01 11:52:00	0.0	NaN	United Kingdom
1937	536545	NaN	1	2010-12-01 14:32:00	0.0	NaN	United Kingdom
1938	536546	NaN	1	2010-12-01 14:33:00	0.0	NaN	United Kingdom
1939	536547	NaN	1	2010-12-01 14:33:00	0.0	NaN	United Kingdom
1940	536549	NaN	1	2010-12-01 14:34:00	0.0	NaN	United Kingdom

```
df = df[df['Price'] > 0]
df[(df.Price==0)].head()
```

BillNo	Itemname	Quantity	Date	Price	CustomerID	Country

Menghapus nilai non produk pada kolom Itemname

```
df.loc[(df['Itemname']=='POSTAGE')|(df['Itemname']=='DOTCOM POSTAGE')|(df['Itemname']=='Adjust bad debt')
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
45	536370	POSTAGE	3	2010-12-01 08:45:00	18.00	12583.0	France
377	536403	POSTAGE	1	2010-12-01 11:27:00	15.00	12791.0	Netherlands
1113	536527	POSTAGE	1	2010-12-01 13:04:00	18.00	12662.0	Germany
1781	536544	DOTCOM POSTAGE	1	2010-12-01 14:32:00	569.77	NaN	United Kingdom
2192	536569	Manual	1	2010-12-01 15:35:00	1.25	16274.0	United Kingdom

```
df=df.loc[(df['Itemname']!='POSTAGE')&(df['Itemname']!='DOTCOM POSTAGE')&(df['Itemname']!='Adjust bad deb
```

Mengisi nilai null dengan nilai ‘-’

```
df.isnull().sum()
```

```
BillNo             0
Itemname           0
Quantity           0
Date               0
Price              0
CustomerID    130813
Country            0
dtype: int64
```

```
df=df.fillna('-')
df.isnull().sum()
```

```
BillNo         0
Itemname       0
Quantity       0
Date           0
Price          0
CustomerID     0
Country        0
dtype: int64
```

Menambahkan kolom baru yaitu Total Price

```
df['Price'] = df['Price'].astype(str).str.replace(',', '.').astype(float)
df['Total price'] = df.Quantity * df.Price
df.head()
```

Pytho

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country	Year	Month	Total price
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010	2.55	17850.0	United Kingdom	2010	12	15.30
1	536365	WHITE METAL LANTERN	6	01.12.2010	3.39	17850.0	United Kingdom	2010	12	20.34
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010	2.75	17850.0	United Kingdom	2010	12	22.00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010	3.39	17850.0	United Kingdom	2010	12	20.34
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010	3.39	17850.0	United Kingdom	2010	12	20.34

Visualisasi 10 Item Terbanyak

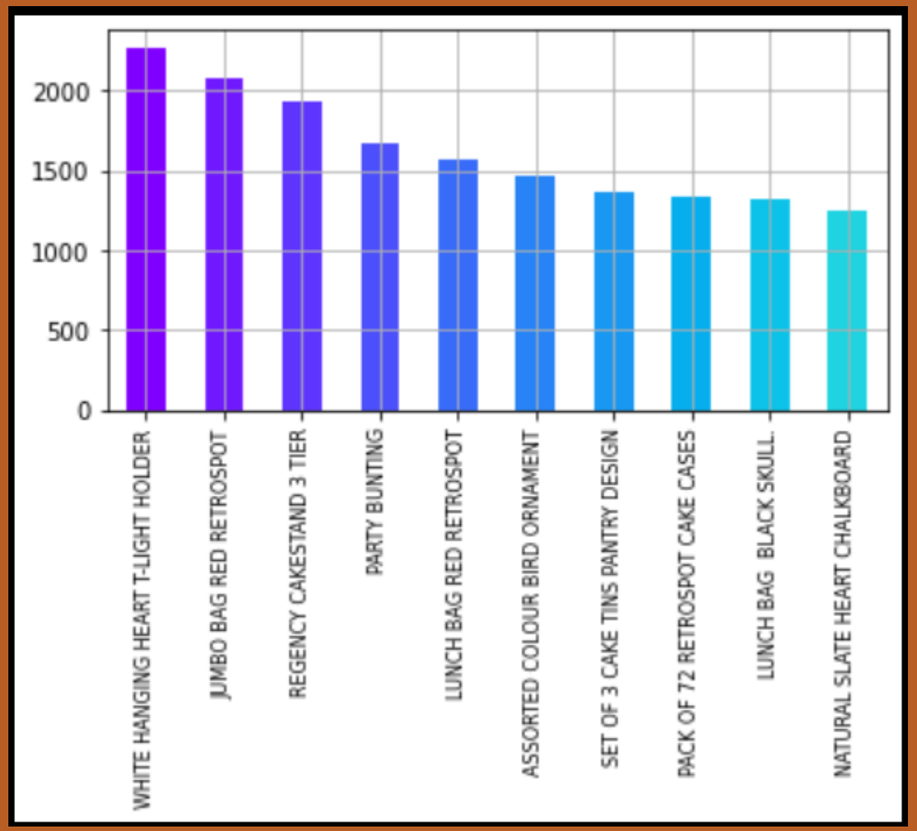
Top 10 highest sales amount items

	Itemname	Price
0	REGENCY CAKESTAND 3 TIER	27103.66
1	AMAZON FEE	13761.09
2	PARTY BUNTING	9741.13
3	SET OF 3 CAKE TINS PANTRY DESIGN	7965.63
4	CREAM SWEETHEART MINI CHEST	7384.51
5	WHITE HANGING HEART T-LIGHT HOLDER	7307.47
6	ENAMEL BREAD BIN CREAM	7041.13
7	SET/4 WHITE RETRO STORAGE CUBES	6861.20
8	RED RETROSPOT CAKE STAND	6668.39
9	IVORY KITCHEN SCALES	6518.76

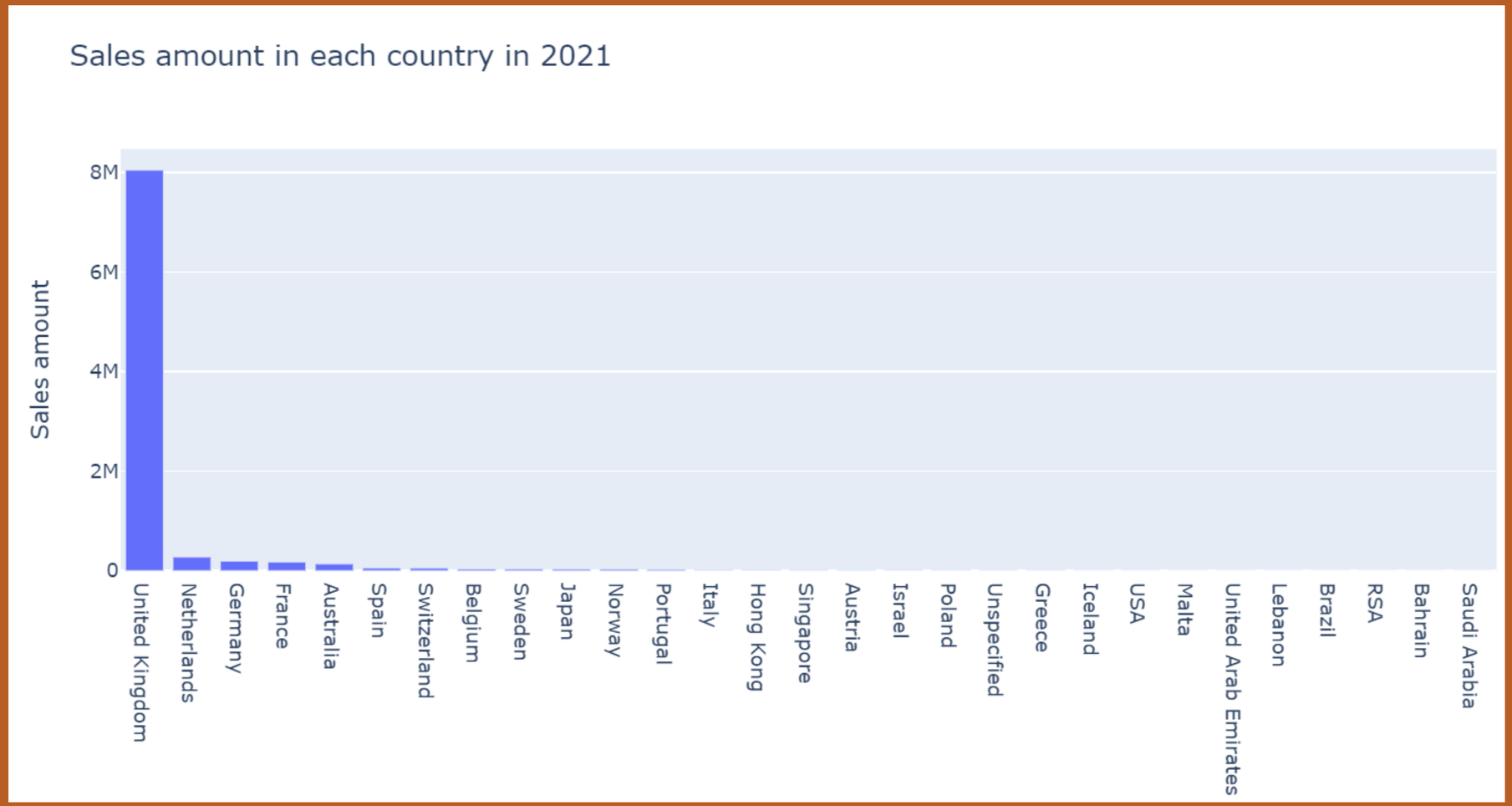
Top 10 most purchased items

	Itemname	Quantity
520583	PAPER CRAFT , LITTLE BIRDIE	80995
59999	MEDIUM CERAMIC TOP STORAGE JAR	74215
405138	WORLD WAR 2 GLIDERS ASSTD DESIGNS	4800
198929	SMALL POPCORN HOLDER	4300
94245	EMPIRE DESIGN ROSETTE	3906
260928	ESSENTIAL BALM 3.5g TIN IN ENVELOPE	3186
51228	FAIRY CAKE FLANNEL ASSORTED COLOUR	3114
154834	FAIRY CAKE FLANNEL ASSORTED COLOUR	3114
416997	SMALL CHINESE STYLE SCISSOR	3000
280572	ASSORTED COLOUR BIRD ORNAMENT	2880

Top 10 most frequently purchased items



Data Visual Jumlah Penjualan di setiap negara periode 2021



Data Modelling

Model Apriori

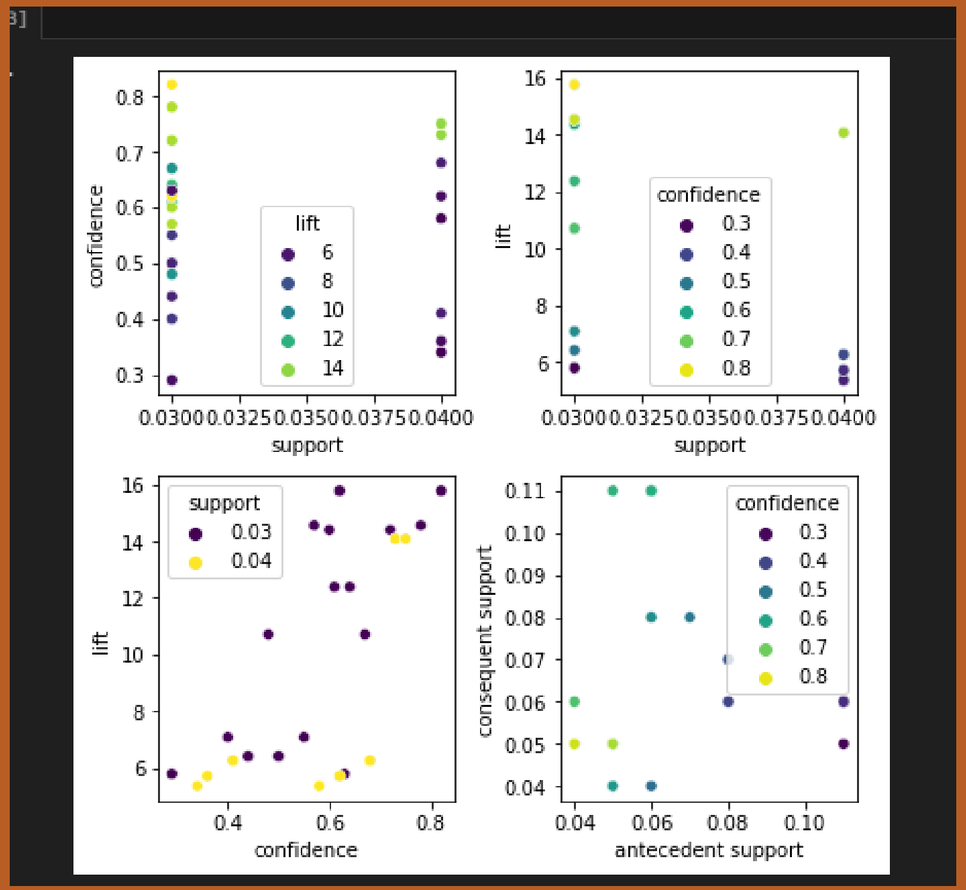
Implementing Apriori

Pada tahap awal, memanggil data yang akan diuji coba dengan melihat nilai Attendence support, confidence dan lift, dll.

```
frequent_itemsets=apriori(basket_sets,min_support=0.03,use_colnames=True)
```

```
rules=round(association_rules(frequent_itemsets,metric='lift',min_threshold=1),2)
rules.head(5)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.05	0.05	0.03	0.61	12.38	0.03	2.41	0.97
1	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.05	0.05	0.03	0.64	12.38	0.03	2.65	0.97
2	(GARDENERS KNEELING PAD CUP OF TEA)	(GARDENERS KNEELING PAD KEEP CALM)	0.04	0.05	0.03	0.72	14.39	0.03	3.41	0.97
3	(GARDENERS KNEELING PAD KEEP CALM)	(GARDENERS KNEELING PAD CUP OF TEA)	0.05	0.04	0.03	0.60	14.39	0.03	2.41	0.98
4	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.05	0.04	0.03	0.62	15.77	0.03	2.52	0.99



Menghitung 5 nilai support terbesar dari Item

```
[ ]  rules[['antecedents','consequents','support']].sort_values('support',ascending=False)[:5].style.background_gradient(cmap=cm).set_precision(2)
```

	antecedents	consequents	support
12	frozenset({'JUMBO SHOPPER VINTAGE RED PAISLEY'})	frozenset({'JUMBO BAG RED RETROSPOT'})	0.04
15	frozenset({'JUMBO BAG RED RETROSPOT'})	frozenset({'JUMBO STORAGE BAG SUKI'})	0.04
6	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.04
7	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.04
10	frozenset({'JUMBO BAG PINK POLKADOT'})	frozenset({'JUMBO BAG RED RETROSPOT'})	0.04

Menghitung 5 nilai Confidence terbesar dari Item

```
rules[['antecedents','consequents','confidence']].sort_values('confidence',ascending=False)[:5].style.background_gradient(cmap=cm).set_precision(2)
```

	antecedents	consequents	confidence
5	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.82
23	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.78
7	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.75
6	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.73
2	frozenset({'GARDENERS KNEELING PAD CUP OF TEA'})	frozenset({'GARDENERS KNEELING PAD KEEP CALM'})	0.72

Menghitung 5 nilai Lift terbesar dari Item

```
[ ]  rules[['antecedents','consequents','lift']].sort_values('lift',ascending=False)[:5].style.background_gradient(cmap=cm).set_precision(2)
```

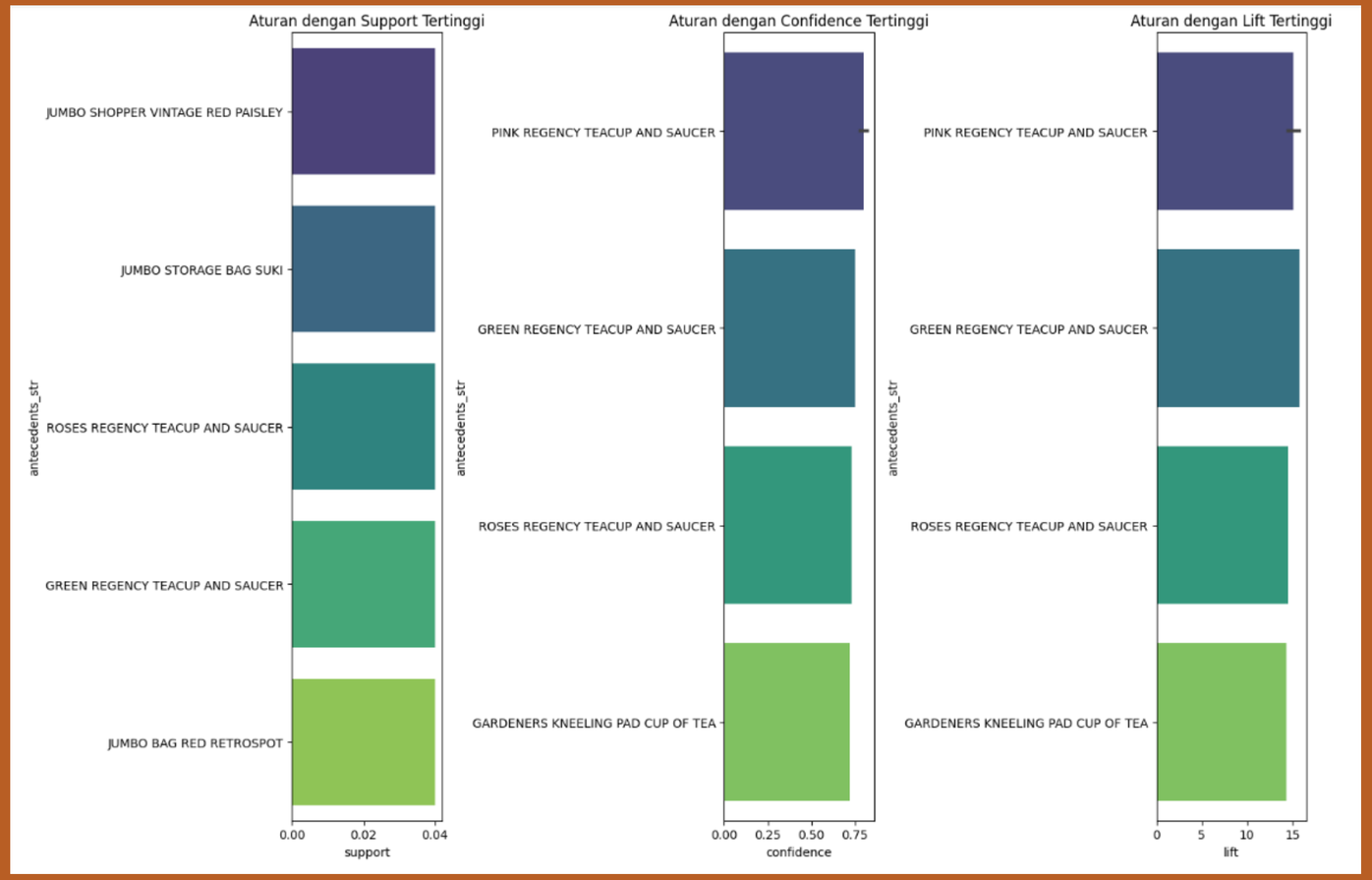
	antecedents	consequents	lift
4	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	15.77
5	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	15.77
23	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	14.55
22	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	14.55
2	frozenset({'GARDENERS KNEELING PAD CUP OF TEA'})	frozenset({'GARDENERS KNEELING PAD KEEP CALM'})	14.39

Menghitung Kombinasi Item terbaik

```
rules[(rules['lift']>=13)&(rules['confidence']>=0.7)].sort_values('lift',ascending=False).style.background_gradient(cmap=cm).set_precision(2)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhang
5	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.04	0.05	0.03	0.82	15.77	0.03	5.28	0.97
23	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.04	0.05	0.03	0.78	14.55	0.03	4.23	0.97
2	frozenset({'GARDENERS KNEELING PAD CUP OF TEA'})	frozenset({'GARDENERS KNEELING PAD KEEP CALM'})	0.04	0.05	0.03	0.72	14.39	0.03	3.41	0.97
6	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.05	0.05	0.04	0.73	14.08	0.04	3.54	0.98
7	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.05	0.05	0.04	0.75	14.08	0.04	3.80	0.98

Visualisasi Hasil akhir



Kesimpulan

Kesimpulan dari analisis :

- **Barang yang paling banyak dibeli adalah GREEN REGENCY TEACUP AND SAUCER**
- **Barang yang paling sering dibeli adalah GREEN REGENCY TEACUP dan PINK REGENCY TEACUP**
- **Item kombinasi pembelian yang terbaik adalah PINK REGENCY TEACUP AND SAUCER dan GREEN REGENCY TEACUP AND SAUCER**

Rekomendasi

PINK/GREEN REGENCY TEACUP AND SAUCER diletakan secara berdekatan dengan item yang jarang dibeli atau diletakan dibelakang sehingga untuk mengambil dapat melewati beberapa item yang jarang dibeli (diberi diskon)

Sekian

dan

Terima Kasih !