

Nama : Dina Cahyo Wibowo
NIM : G.231.22.0004
Matkul : Data Mining

```
[2] 1 import pandas as pd
2 from mlxtend.frequent_patterns import apriori
3 from mlxtend.frequent_patterns import association_rules
4
5 df = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx')
6 df.head()
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transform_cell` manually and `should_run_async` (code)

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Kode di atas ketika dijalankan akan mengimport beberapa library dan juga mengimport file excel dari link yang sudah ada di atas

```
[3] 1 df['Description'] = df['Description'].str.strip()
2 df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
3 df['InvoiceNo'] = df['InvoiceNo'].astype('str')
4 df = df[~df['InvoiceNo'].str.contains('C')]
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transform_cell` manually and `should_run_async` (code)

kode ini berguna untuk membersihkan data mulai dari Menghapus spasi di awal dan akhir teks pada kolom Description dan menghapus baris yang memiliki nilai NaN pada kolom InvoiceNo

```
[4] 1 basket = (df[df['Country'] == "France"]
2           .groupby(['InvoiceNo', 'Description'])['Quantity']
3           .sum().unstack().reset_index().fillna(0)
4           .set_index('InvoiceNo'))
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transform_cell` manually and `should_run_async` (code)

sedangkan untuk kode ini berfungsi untuk menyiapkan data mulai dari Menyaring transaksi untuk negara Prancis. Mengelompokkan data berdasarkan nomor faktur dan deskripsi produk, kemudian menjumlahkan kuantitas produk. Mengubah bentuk data dengan unstack sehingga setiap produk menjadi kolom. Mengatur ulang indeks dan mengisi nilai NaN dengan 0.

```
[5] 1 def encode_units(x):
2     if x <= 0:
3         return 0
4     if x >= 1:
5         return 1
6
7 basket_sets = basket.applymap(encode_units)
8 basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transform_cell` manually and `should_run_async` (code)

Kode ini berfungsi untuk Mendefinisikan fungsi encode_units untuk mengubah kuantitas produk menjadi 0 atau 1. Menerapkan fungsi encode_units ke semua data pada basket. Menghapus kolom 'POSTAGE' dari basket_sets

```
[6] 1 frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/mlxtend/frequent_patterns/fpcommon.py:110: D
```

```
[7] 1 rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
2 rules.head()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the
and should_run_async(code)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.102041	0.096939	0.073980	0.725000	7.478947	0.064088	3.283859	0.964734
1	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.096939	0.102041	0.073980	0.763158	7.478947	0.064088	3.791383	0.959283
2	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181	0.979224
3	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878	0.976465
4	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE RED)	0.102041	0.094388	0.073980	0.725000	7.681081	0.064348	3.293135	0.968652

Kedua kolom code di atas berfungsi untuk Mengimpor fungsi `apriori` dan `association_rules` dari pustaka `mlxtend`. Menemukan kumpulan item yang sering muncul dengan `apriori` dan minimum support 0.07. Menghasilkan aturan asosiasi dengan `association_rules` berdasarkan metrik "lift" dan ambang batas minimal 1. Dan juga menampilkan lima aturan asosiasi pertama.

```
[8] 1 rules[ (rules['lift'] >= 6) &
2 (rules['confidence'] >= 0.8) ]
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result t
and should_run_async(code)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
2	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181	0.979224
3	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878	0.976465
17	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.127551	0.132653	0.102041	0.800000	6.030769	0.085121	4.336735	0.956140
18	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.137755	0.127551	0.122449	0.888889	6.968889	0.104878	7.852041	0.993343
19	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.127551	0.137755	0.122449	0.960000	6.968889	0.104878	21.556122	0.981725
20	(SET/6 RED SPOTTY PAPER CUPS, SET/20 RED RETRO...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.127551	0.099490	0.975000	7.644000	0.086474	34.897959	0.967949
21	(SET/6 RED SPOTTY PAPER CUPS, SET/6 RED SPOTTY...	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.122449	0.132653	0.099490	0.812500	6.125000	0.083247	4.625850	0.953488
22	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.137755	0.099490	0.975000	7.077778	0.085433	34.489796	0.956294

Sedangkan kode di atas ini berfungsi untuk Menyaring aturan asosiasi yang memiliki nilai lift minimal 6 dan confidence minimal 0.8. Serta menampilkan aturan yang memenuhi kriteria tersebut

```
[9] 1 basket['ALARM CLOCK BAKELIKE GREEN'].sum()
2
3 340.0
4
5 basket['ALARM CLOCK BAKELIKE RED'].sum()
6
7 316.0
```


```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: and should_run_async(code)
```

Mengakses kolom `ALARM CLOCK BAKELIKE GREEN` dan `ALARM CLOCK BAKELIKE GREEN` dalam dataframe `basket`. Serta Menghitung total kuantitas untuk produk dengan menggunakan fungsi `sum()`. dan menunjukkan total unit yang terjual untuk produk tersebut. Hasilnya adalah 340.0. dan 316.0

```

1 basket2 = (df[df['Country'] == "Germany"]
2            .groupby(['InvoiceNo', 'Description'])['Quantity']
3            .sum().unstack().reset_index().fillna(0)
4            .set_index('InvoiceNo'))
5
6 basket_sets2 = basket2.applymap(encode_units)
7 basket_sets2.drop('POSTAGE', inplace=True, axis=1)
8 frequent_itemsets2 = apriori(basket_sets2, min_support=0.05, use_colnames=True)
9 rules2 = association_rules(frequent_itemsets2, metric="lift", min_threshold=1)
10
11 rules2[ (rules2['lift'] >= 4) &
12         (rules2['confidence'] >= 0.5)]

```

 /usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: and should_run_async(code)
 /usr/local/lib/python3.10/dist-packages/mlxtend/frequent_patterns/fpcommon.py:110: D
 warnings.warn(

Sedangkan untuk kode di atas berguna untuk menspesifikkan transaksi yang hanya berasal dari Jerman. Mengelompokkan data berdasarkan `InvoiceNo` dan `Description`, lalu menjumlahkan kuantitas produk. Mengubah bentuk data dengan `unstack` sehingga setiap produk menjadi kolom. Serta mengatur ulang indeks dan mengisi nilai `NaN` dengan 0. dan juga menerapkan fungsi `encode_units` untuk mengubah kuantitas produk menjadi 0 atau 1 (biner). Sedangkan sisanya kurang lebih sama seperti kode kode sebelumnya