

Nama : Victor Divo Mahendra
NIM : G.231.22.0083

Penjelasan kode

```
# Melakukan import package yang akan digunakan
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# Baca data dari excel yang ada di url dibawah
df =
pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx')
df.head()
```

Kode diatas digunakan untuk mengimport package dan melakukan load data awal dari excel dengan url diatas

```
# Melakukan preprocessing pada data, melakukan data clean up
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]

# Melakukan rename kolom
df.rename(columns={'Description': 'Deskripsi', 'InvoiceNo': 'NomorInvoice'}, inplace=True)
df.head()
```

Melakukan preprocessing pada data seperti menghapus spasi yang tidak terpakai pada kolom deskripsi, menghapus data yang memiliki Nilai Kosong pada bagian Nomor Invoice, mengubah tipe data dari Nomor Invoice dan melakukan filter Nomor Invoice yang tidak memiliki nilai C didalamnya

```
# mengambil data basket dengan negara France dan melakukan groupby Nomor Invoice dan Deskripsi
basket = (df[df['Country'] == "France"]
          .groupby(['NomorInvoice', 'Deskripsi'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('NomorInvoice'))
```

Melakukan proses filtering, mengambil data dari france dan melakukan group by nomor invoice dan deskripsi, lalu mengambil quantity dengan index yang dipakai ada nomor invoice, dan mengisi data yang kosong dengan 0

```
# membuat fungsi untuk setiap row, mengembalikan data apakah sebuah barang ada atau tidak (0/1)

def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

membuat fungsi untuk setiap row, mengembalikan data apakah sebuah barang ada atau tidak dan menghapus kolom postage

```
# Melakukan analisa asosiasi dengan algoritma apriori dengan aturan tertentu
frequent_itemsets = apriori(basket_sets, min_support=0.07,
                             use_colnames=True)
```

Melakukan analisa asosiasi dengan algoritma apriori dengan aturan tertentu

```
# Mengambil hasil dari analisa dengan aturan yang sudah ditetapkan sebelumnya
rules = association_rules(frequent_itemsets, metric="lift",
                           min_threshold=1)
rules.head()
```

Mengambil hasil dari analisa dengan aturan yang sudah ditetapkan sebelumnya dan menampilkannya ke output

```
# Melakukan filter/menampilkan hasil barang yang memiliki lift >= 6 dan
memiliki confidence >= 0.8
rules[ (rules['lift'] >= 6) &
        (rules['confidence'] >= 0.8) ]
```

Hanya menampilkan barang yang memiliki aturan tertentu seperti diatas

```
# Melihat popularitas dari jam hijau dan jam merah
print(basket['ALARM CLOCK BAKELIKE GREEN'].sum())

print(basket['ALARM CLOCK BAKELIKE RED'].sum())
```

Melihat banyak data dari jam hijau dan merah

```
# mengambil data basket dengan negara Germany dan melakukan groupby Nomor
Invoice dan Deskripsi
basket2 = (df[df['Country'] == "Germany"]
           .groupby(['NomorInvoice', 'Deskripsi'])['Quantity']
           .sum().unstack().reset_index().fillna(0)
           .set_index('NomorInvoice'))

# Proses yang sama dengan sebelumnya dengan data dari Negara France
basket_sets2 = basket2.applymap(encode_units)
basket_sets2.drop('POSTAGE', inplace=True, axis=1)
frequent_itemsets2 = apriori(basket_sets2, min_support=0.05,
                              use_colnames=True)
rules2 = association_rules(frequent_itemsets2, metric="lift",
                           min_threshold=1)

# Hanya menampilkan hasil yang memiliki lift >= 4 dan confidence >= 0.5
rules2[ (rules2['lift'] >= 4) &
        (rules2['confidence'] >= 0.5)]
```

Proses yang sama seperti sebelumnya, namun menggunakan data dari negara Germany