

NAMA : Naufal Hanif Rajendra

NIM : G.231.22.0125

## **Penjelasan Kode Langkah demi Langkah Praktikum 4**

### **1. Memuat Data (Langkah 5-6)**

Langkah pertama adalah memuat data penjualan online dari file Excel. Data tersebut disimpan dalam variabel df.

Python

```
df = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-  
databases/00352/Online2@Retail.xlsx')  
  
df.head()
```

### **2. Pembersihan Data (Langkah 1-2, 4)**

Langkah selanjutnya adalah membersihkan data sebelum digunakan untuk analisis. Dalam langkah ini, dilakukan beberapa hal berikut:

- Menghilangkan spasi di kolom Description (langkah 1).
- Menghapus baris yang memiliki nilai InvoiceNo kosong (langkah 2).
- Mengubah tipe data kolom InvoiceNo menjadi string (langkah 3).
- Menghapus baris yang memiliki nilai InvoiceNo mengandung huruf C (langkah 4).

Python

```
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')]
```

### 3. Analisis Market Basket (Langkah 3)

Langkah terakhir adalah melakukan analisis market basket untuk menemukan pola pembelian produk yang sering dibeli bersama oleh pelanggan. Dalam langkah ini, dilakukan beberapa hal berikut:

- Mengelompokkan data berdasarkan negara pelanggan (langkah 1).
- Menghitung jumlah pembelian setiap produk untuk setiap kelompok pelanggan (langkah 2).
- Mengubah data tersebut menjadi format tabel (langkah 3).
- Mengatur indeks tabel menjadi kolom InvoiceNo (langkah 4).

Python

```
basket = (df[df['Country'] == "France"]
          .groupby(['InvoiceNo', 'Description'])['quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

### 1. Memuat Data

Langkah pertama adalah memuat data penjualan online dari file Excel. Data tersebut disimpan dalam variabel `df`.

Python

```
import pandas as pd

df = pd.read_excel('Online2@Retail.xlsx')
df.head()
```

### 2. Pembersihan Data

Langkah selanjutnya adalah membersihkan data sebelum digunakan untuk analisis. Dalam langkah ini, dilakukan beberapa hal berikut:

- Menghilangkan spasi di kolom `Description`.
- Menghapus baris yang memiliki nilai `InvoiceNo` kosong.
- Mengubah tipe data kolom `InvoiceNo` menjadi string.
- Menghapus baris yang memiliki nilai `InvoiceNo` mengandung huruf C.

Python

```
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')]
```

### 3. Analisis Market Basket

Langkah terakhir adalah melakukan analisis market basket untuk menemukan pola pembelian produk yang sering dibeli bersama oleh pelanggan. Dalam langkah ini, dilakukan beberapa hal berikut:

- Mengelompokkan data berdasarkan negara pelanggan.
- Menghitung jumlah pembelian setiap produk untuk setiap kelompok pelanggan.
- Mengubah data tersebut menjadi format tabel.
- Mengatur indeks tabel menjadi kolom `InvoiceNo`.

Python

```
from mlxtend.frequent_itemsets import apriori, association_rules

frequent_itemsets = apriori(basket, min_support=0.07, use_colnames=True)
association_rules = association_rules(frequent_itemsets, metric="lift",
min_threshold=1)
```

### 4. Menampilkan Hasil Analisis

Langkah terakhir adalah menampilkan hasil analisis market basket. Dalam langkah ini, dilakukan beberapa hal berikut:

- Menampilkan tabel `association_rules` yang berisi informasi tentang pola pembelian produk yang sering dibeli bersama.
- Menampilkan 2 baris pertama dari tabel `association_rules`.

Python

```
print(association_rules.head(2))
```

### 1. Memuat Data

Langkah pertama dalam kode ini adalah memuat data transaksi penjualan dari file Excel. Ini dilakukan dengan menggunakan `pandas` perpustakaan.

ular piton

```
import pandas as pd

df = pd.read_excel('Online2@Retail.xlsx')
df.head()
```

### 2. Pembersihan Data

Sebelum melakukan analisis, penting untuk membersihkan data untuk memastikan keakuratan dan konsistensinya. Dalam hal ini, langkah-langkah pembersihan data berikut dilakukan:

- Hapus spasi tambahan dari `Description` kolom.
- Jatuhkan baris dengan nilai yang hilang di `InvoiceNo` kolom.
- Ubah `InvoiceNo` kolom menjadi tipe string.
- Jatuhkan baris yang `InvoiceNo` nilainya berisi huruf C.

ular piton

```
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')]
```

### 3. Analisis Keranjang Pasar

Inti dari analisis keranjang pasar dilakukan dengan menggunakan `mlxtend` perpustakaan. Langkah-langkah berikut terlibat:

- **Mengelompokkan data berdasarkan negara:** Data pertama kali dikelompokkan berdasarkan negara pelanggan. Hal ini memungkinkan untuk mengidentifikasi pola yang spesifik untuk setiap negara.
- **Menghitung jumlah produk:** Untuk setiap kelompok transaksi, kuantitas setiap produk dihitung. Ini memberikan hitungan berapa kali setiap produk dibeli di setiap negara.
- **Mengonversi data ke format tabel:** Data kuantitas produk diubah menjadi format tabel, di mana setiap baris mewakili negara dan setiap kolom mewakili produk. Nilai dalam tabel mewakili kuantitas setiap produk yang dibeli di negara tersebut.
- **Pengaturan indeks:** Indeks tabel diatur ke `InvoiceNo` kolom. Hal ini memungkinkan identifikasi transaksi individu menjadi lebih mudah.

ular piton

```
from mlxtend.frequent_itemsets import apriori, association_rules

basket = (df[df['Country'] == "France"]
          .groupby(['InvoiceNo', 'Description'])['quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

### 4. Menghasilkan Itemset yang Sering

Kumpulan item yang sering adalah kelompok produk yang sering dibeli bersama. Kumpulan item ini diidentifikasi menggunakan `apriori()` fungsi. Parameter `min_support` menentukan jumlah minimum transaksi di mana suatu itemset harus dianggap sering terjadi

ular piton.

```
frequent_itemsets = apriori(basket, min_support=0.07, use_colnames=True)
```

### 5. Menghasilkan Peraturan Asosiasi

Aturan asosiasi adalah hubungan antar itemset, yang menunjukkan bahwa kehadiran satu itemset (anteseden) meningkatkan kemungkinan kehadiran itemset lain (konsekuen). Aturan-aturan ini dihasilkan menggunakan `association_rules()` fungsi tersebut. Parameter `metric` menentukan metrik yang akan digunakan untuk mengevaluasi kekuatan aturan. Parameter `min_threshold` menentukan nilai ambang batas minimum untuk metrik.

ular piton

```
association_rules = association_rules(frequent_itemsets, metric="lift",
```