

Nama : Afef Dwi Hidayat

Nim : G.231.22.0075

Data Mining

Program : <https://github.com/AfiefDH/MarketAnalysis.git>

Penjelasan Program Market Basket Analisis

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

df = pd.read_excel('https://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx')

df.rename(columns={
    'InvoiceNo': 'NomorFaktur',
    'StockCode': 'KodeStok',
    'Description': 'Deskripsi',
    'Quantity': 'Kuantitas',
    'InvoiceDate': 'TanggalFaktur',
    'UnitPrice': 'HargaSatuan',
    'CustomerID': 'IDPelanggan',
    'Country': 'Negara'
}, inplace=True)

df.head()
```

Dimulai dengan mengimport pustaka 'pandas' untuk manipulasi data dan 'mlxtend' untuk algoritma dan aturan asosiasi

Kemudian data diambil dari URL yang diberikan dan dimuat ke dalam dataframe 'df', disini juga terdapat 'rename' untuk mengubah nama kolom ke bahasa indonesia, dan 'head()' digunakan untuk menampilkan 5 baris pertama

```
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')]
```

Dibagian ini melakukan Data Cleaning untuk :

- Menghilangkan spasi di awal dan akhir dari kolom 'Description'
- Menghapus baris dengan nilai 'InvoiceNo' yang kosong
- Mengubah tipe data 'InvoiceNo' menjadi string
- Menghapus faktur yang mengandung 'C' ( biasanya menandakan transaksi yang dibatalkan )

```

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

```

Memfilter bagian data Transformation dari Prancis, dan mengelompokkan data berdasarkan 'InvoiceNo' dan 'Description' lalu menjumlahkan 'Quantity', sehingga mengubah data 'Description' menjadi kolom dan 'InvoiceNo' menjadi Indeks

```

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)

```

Encoding Data , Fungsi 'encode\_units' mengubah nilai kuantitas menjadi biner ( 1 jika ada transaksi, 0 Jika tidak ada ), 'applymap' digunakan untuk menerapkan fungsi ini ke seluruh elemen dataframe 'basket\_sets', dan menghapus kolom 'POSTAGE' dari data

```

frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)

```

Applying Apriori Algorithm : Menggunakan algoritma Apriori untuk menemukan itemset yang sering muncul dengan dukungan minimal 7%

```

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head()

```

Generating Association Rules : Menghasilkan aturan asosiasi dari itemset yang sering muncul dengan menggunakan metrik 'lift' dan ambang batas minimal 1 , dan menampilkan 5 aturan pertama

```

rules[ (rules['lift'] >= 6) &
       (rules['confidence'] >= 0.8)]

```

Memfilter Peraturan : Menyaring aturan yang memiliki 'lift' minimal 6 dan 'confidence' minimal 80%

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

```
340.0
```

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

```
316.0
```

Menjumlah kuantitas untuk item 'ALARM CLOCK BAKELIKE GREEN' dan 'ALARM CLOCK BAKELIKE RED'

```
basket2 = (df[df['Country'] == "Germany"]
           .groupby(['InvoiceNo', 'Description'])['Quantity']
           .sum().unstack().reset_index().fillna(0)
           .set_index('InvoiceNo'))

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)

rules2[ (rules2['lift'] >= 4) &
        (rules2['confidence'] >= 0.5)]
```

Langkah-langkah yang sama seperti untuk yang Prancis, tetapi sekarang diganti pada data untuk Jerman

Menyaring aturan yang memiliki 'lift' minimal 4 dan 'confidence' minimal 50% untuk Jerman