

# MODUL

# STMIK WIDYA PRATAMA



# DATA MINING

## Asosiasi



## **DESKRIPSI MATAKULIAH**

## CAPAIAN PEMBELAJARAN

## **DOSEN PENGAMPU**

# **SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

## **(STMIK) WIDYA PRATAMA**



## CAPAIAN PEMBELAJARAN

## MATERI PEMBELAJARAN

## 1. Perhitungan Python Apriory

Adapun dataset yang digunakan adalah GroceryStoreDataset.csv. berikut perintah nya :

```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules
```

Keterangan kode :

- Import pandas as pd biasa digunakan untuk mengubah dimensi data, membuat tabel, memeriksa data, membaca data dan lain sebagainya.
- Import numpy as np berfungsi untuk memudahkan operasi perhitungan tipe data numeric seperti penjumlahan, perkalian, pengurangan, pemangkatan dan operasi aritmatika lainnya
- From mlxtend.frequent\_patterns import apriori, association\_rules memanggil library mlxtend untuk melakukan asosiasi menggunakan algoritma apriori

```
df = pd.read_csv('GroceryStoreDataSet.csv', names = ['products'], sep = ',')
df.head()
```

	products
0	MILK,BREAD,BISCUIT
1	BREAD,MILK,BISCUIT,CORNFLAKES
2	BREAD,TEA,BOURNVITA
3	JAM,MAGGI,BREAD,MILK
4	MAGGI,TEA,BISCUIT

Keterangan kode :

- Df = pd.read\_csv('GroceryDataset.csv', names = ['products'], sep = ',') untuk membaca dataset yang sudah disediakan dalam format csv dengan nama variable "products"
- Head() digunakan untuk menampilkan sebanyak 5 data teratas.

```
df.shape
```

```
(20, 1)
```

Keterangan kode :

- df.shape untuk menampilkan kumpulan data

```
data = list(df["products"].apply(lambda x:x.split(",") ))  
  
[['MILK', 'BREAD', 'BISCUIT'],  
 ['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],  
 ['BREAD', 'TEA', 'BOURNVITA'],  
 ['JAM', 'MAGGI', 'BREAD', 'MILK'],  
 ['MAGGI', 'TEA', 'BISCUIT'],  
 ['BREAD', 'TEA', 'BOURNVITA'],  
 ['MAGGI', 'TEA', 'CORNFLAKES'],  
 ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],  
 ['JAM', 'MAGGI', 'BREAD', 'TEA'],  
 ['BREAD', 'MILK'],  
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],  
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],  
 ['COFFEE', 'SUGER', 'BOURNVITA'],  
 ['BREAD', 'COFFEE', 'COCK'],  
 ['BREAD', 'SUGER', 'BISCUIT'],  
 ['COFFEE', 'SUGER', 'CORNFLAKES'],  
 ['BREAD', 'SUGER', 'BOURNVITA'],  
 ['BREAD', 'COFFEE', 'SUGER'],  
 ['BREAD', 'COFFEE', 'SUGER'],  
 ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

Keterangan kode :

- data = list(df[“products”.apply(lambda x:x.split(“,”) )]) untuk mengambil data yang terdapat di dalam kolom produk

```
from mlxtend.preprocessing import TransactionEncoder  
a = TransactionEncoder()  
a_data = a.fit(data).transform(data)  
df = pd.DataFrame(a_data,columns=a.columns_)  
df = df.replace(False,0)  
df
```

Keterangan kode :

- from mlxtend.preprocessing import TransactionEncoder, untuk memanggil

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	SUGER	TEA
0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
1	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
2	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
3	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
4	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0
5	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
6	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
7	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0
8	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0
9	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
10	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
11	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
12	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
13	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
14	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
15	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0
16	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
17	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
18	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
19	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0

```
df = apriori(df, min_support = 0.2, use_colnames = True, verbose = 1)
df
```

Keterangan kode :

- df = apriori(df, min\_support = 0.2, use\_colnames= True, verbose = 1) untuk menampilkan gambar sebagai berikut

Processing 42 combinations | Sampling itemset size 3

	support	itemsets
0	0.35	(BISCUIT)
1	0.20	(BOURNVITA)
2	0.65	(BREAD)
3	0.40	(COFFEE)
4	0.30	(CORNFLAKES)
5	0.25	(MAGGI)
6	0.25	(MILK)
7	0.30	(SUGER)
8	0.35	(TEA)
9	0.20	(BREAD, BISCUIT)
10	0.20	(BREAD, MILK)
11	0.20	(SUGER, BREAD)
12	0.20	(BREAD, TEA)
13	0.20	(CORNFLAKES, COFFEE)
14	0.20	(SUGER, COFFEE)
15	0.20	(MAGGI, TEA)

Untuk

```
# fungsi aturan Asosiasi
df_ar = association_rules(df, metric = "confidence", min_threshold = 0.6)
df_ar
```

Keterangan kode :

- df\_ar = association\_rules(df., metric = “confidence”, min\_threshold = 0.6) untuk menampilkan gambar sebagai berikut

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	0.0375	1.75
1	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	0.0050	1.05
2	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80
3	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80
4	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	0.1125	3.25

Berdasarkan perhitungan menggunakan python di atas, pembeli yang membeli gula atau “SUGAR” kemungkinan akan membeli roti atau “BREAD” lebih banyak dibanding pembeli yang tidak membeli gula.

## TES FORMATIF