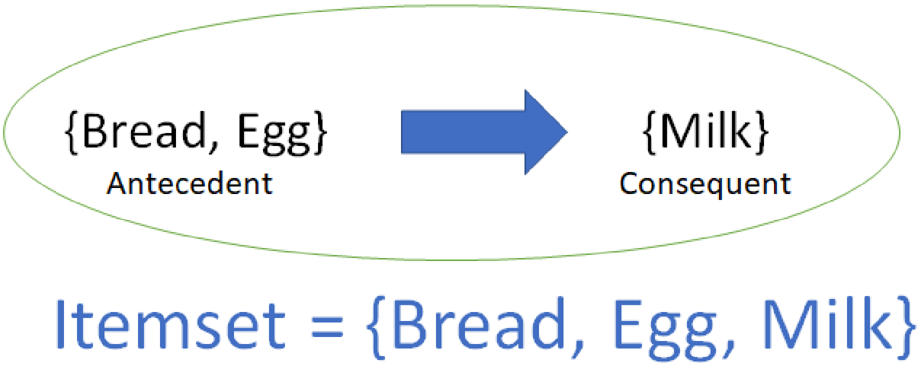
Pertemuan 4: Market Basket Analysis

Association Rule

Market Basket Analysis bertujuan menemukan pola-pola hubungan yang sering muncul di dalam data, misalnya data transaksi penjualan. Pola yang dicari inilah yang disebut dengan Association Rule, atau bisa diterjemahkan bebas sebagai hubungan (asosiasi) antara kombinasi beberapa item (barang, orang, produk, atau apapun yang diwakili oleh kata benda) yang sering muncul bersamaan.

Sebuah rule memiliki antecedent (hal yang mengawali) yang ditulis di kiri tanda panah, dan sebuah consequent (efek atau akibat) sebelah kanan anak panah.



Nilai-nilai yang ada di dalam tanda kurung kurawal {} disebut itemset, bisa terdiri dari satu item atau lebih. Rule diatas menunjukkan bahwa jika roti dan/atau telur dibeli, maka susu (kemungkinan besar) akan dibeli. Secara teoritis artinya telah terbentuk suatu aturan, yaitu kemunculan {roti,telur} mengakibatkan kemunculan susu.

Note: Rule ini berlaku satu arah, artinya ketika membeli susu, maka {roti,telur} belum tentu akan dibeli.

Association rule tidah hanya bisa dimanfaatkan untuk Market basket analysis. Banyak implementasi lain di industri keuangan misalnya untuk menemukan transaksi kartu kredit yang tidak wajar atau di bidang penelitian ilmiah misalnya untuk mencari pola dalam DNA manusia.

Sebelum mencari pola-pola hubungan antar-itemset, terlebih dahulu membuang itemset yang tidak akan pernah berhubungan satu sama lain. Misalkan rule {roti, detergen} adalah rule yang tidak akan terjadi, atau amat jarang terjadi, tentunya dengan asumsi bahwa seseorang akan membeli sarapan pagi.

Produk diatas tidak akan memberikan kontribusi pada proses analisis, sehingga perlu dikeluarkan dari dataset untuk mencegah komputer menangani data yang terlalu besar. Bila sebuah toko memiliki 10 macam produk, maka diperlukan 2^{10}= 1.024 kombinasi itemset.

Salah satu algoritma yang populer untuk memilih data bagi keperluan Market Basket Analysis adalah Apriori. Prinsip dasar apriori adalah meyakini bahwa semua subset (sebagian) dari sebuah itemset harus memiliki frekuensi kemunculan yang tinggi.

In [ ]:
```
pip install mlxtend
```

```
Collecting mlxtend
  Downloading mlxtend-0.21.0-py2.py3-none-any.whl (1.3 MB)
     |████████████████████████████████| 1.3 MB 1.4 MB/s eta 0:00:01
Requirement already satisfied: joblib>=0.13.2 in /opt/anaconda3/lib/python
3.9/site-packages (from mlxtend) (1.1.0)
Requirement already satisfied: matplotlib>=3.0.0 in /opt/anaconda3/lib/pyth
on3.9/site-packages (from mlxtend) (3.5.1)
Requirement already satisfied: scipy>=1.2.1 in /opt/anaconda3/lib/python3.
9/site-packages (from mlxtend) (1.7.3)
Requirement already satisfied: pandas>=0.24.2 in /opt/anaconda3/lib/python
3.9/site-packages (from mlxtend) (1.4.2)
Requirement already satisfied: setuptools in /opt/anaconda3/lib/python3.9/s
ite-packages (from mlxtend) (61.2.0)
Requirement already satisfied: scikit-learn>=1.0.2 in /opt/anaconda3/lib/py
thon3.9/site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: numpy>=1.16.2 in /opt/anaconda3/lib/python3.
9/site-packages (from mlxtend) (1.21.5)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/anaconda3/lib/pyth
on3.9/site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/lib/python3.
9/site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/anaconda3/lib/pytho
n3.9/site-packages (from matplotlib>=3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in /opt/anaconda3/lib/python3.
9/site-packages (from matplotlib>=3.0.0->mlxtend) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/lib/p
ython3.9/site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/lib/pyth
on3.9/site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python
3.9/site-packages (from matplotlib>=3.0.0->mlxtend) (21.3)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.
9/site-packages (from pandas>=0.24.2->mlxtend) (2021.3)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.9/sit
e-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/anaconda3/lib/p
ython3.9/site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
Installing collected packages: mlxtend
Successfully installed mlxtend-0.21.0
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import mlxtend as ml
# from scipy import sparse
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import fpgrowth
```

In [ ]:
```
df0 = pd.read_csv("/Users/dafinazwa/Downloads/semua_transaksi_ch11b.csv")
df0
```

Out[ ]:

	TRX_ID	TRX_TS	PRODUCT_ID	PRODUCT_NAME	SALES
0	85	05/31/17 21.14	263	Cappuccino	34000
1	85	05/31/17 21.14	227	Kopi Tubruk	44000
2	85	05/31/17 21.14	268	Macchiato	26000
3	85	05/31/17 21.14	268	Macchiato	26000
4	85	05/31/17 21.14	268	Macchiato	26000
...	...	...	...	...	...
30498	8850	08/30/19 19.03	233	Rawon	36000
30499	8851	08/26/19 13.28	236	Nasi Putih	7000
30500	8851	08/30/19 19.03	233	Rawon	36000
30501	8852	08/26/19 13.28	236	Nasi Putih	7000
30502	8852	08/30/19 19.03	233	Rawon	36000

30503 rows × 5 columns

In [ ]:

```
df0.index
```

Out[ ]:

```
RangeIndex(start=0, stop=30503, step=1)
```

In [ ]:

```
df = pd.read_csv("/Users/dafinazwa/Downloads/semua_transaksi_ch11b.csv",pars
df
```

Out[ ]:

	TRX_TS	PRODUCT_ID	PRODUCT_NAME	SALES
TRX_ID				
85	05/31/17 21.14	263	Cappuccino	34000
85	05/31/17 21.14	227	Kopi Tubruk	44000
85	05/31/17 21.14	268	Macchiato	26000
85	05/31/17 21.14	268	Macchiato	26000
85	05/31/17 21.14	268	Macchiato	26000
...	...	...	...	...
8850	08/30/19 19.03	233	Rawon	36000
8851	08/26/19 13.28	236	Nasi Putih	7000
8851	08/30/19 19.03	233	Rawon	36000
8852	08/26/19 13.28	236	Nasi Putih	7000
8852	08/30/19 19.03	233	Rawon	36000

30503 rows × 4 columns

In [ ]:

```
df.index
```

Out[ ]:

```
Int64Index([  85,   85,   85,   85,   85,   85,   85,   85,   86,   86,
            ...
            8848, 8848, 8849, 8849, 8850, 8850, 8851, 8851, 8852, 8852],
           dtype='int64', name='TRX_ID', length=30503)
```

Data Profiling

In [ ]:
```
df['PRODUCT_NAME'].value_counts()
```

Out[ ]:
```
Teh Tubruk                       4677
Mineral Water                    2736
Pisang Goreng                    2206
Nasi Putih                       2016
Nasi Goreng Jawa                 1810
Kopi Tubruk                      1799
Soto Ayam                        1758
Extra Kerupuk Putih              1567
Rawon                            1441
Bakmi Godog Keju                 1415
Bakmi Goreng Jawa                1184
Fresh Orange Juice               1163
Latte                            1077
Teh Tarik                         957
Roti Bakar                        854
Cappuccino                        811
Macchiato                         634
Hot/Ice Lemon Tea                 599
Americano                         506
French Fries                      486
Spaghetti Bolognese               303
Espresso                          111
Macchiato                         100
Iced Tea                           87
Milo Dinosaur                      43
Bakwan Goreng                      32
Double Espresso                    22
Hot Chocolate                      18
Indomie Goreng (R)                 18
Green Tea                          15
Indomie Goreng (J)                 13
Telur Orak Arik                     9
Pisang Goreng Tuna Sandwich         7
Es Teh Susu (Gula Jawa)             5
Es Batu                             5
Lemon Tea                           5
Jasmine                             4
Iced Peach Tea                      4
Indomie Goreng Aceh                 4
Takjil                              2
Name: PRODUCT_NAME, dtype: int64
```

In [ ]:
```
df1 = df.drop(['PRODUCT_ID'],axis=1).groupby('PRODUCT_NAME').sum()
df1
```

```
/var/folders/zr/wxf80hrn0110_vv7xq21w5n80000gn/T/ipykernel_1865/961606918.p
y:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.s
um is deprecated. In a future version, numeric_only will default to False.
Either specify numeric_only or select only columns which should be valid fo
r the function.
  df1 = df.drop(['PRODUCT_ID'],axis=1).groupby('PRODUCT_NAME').sum()
```

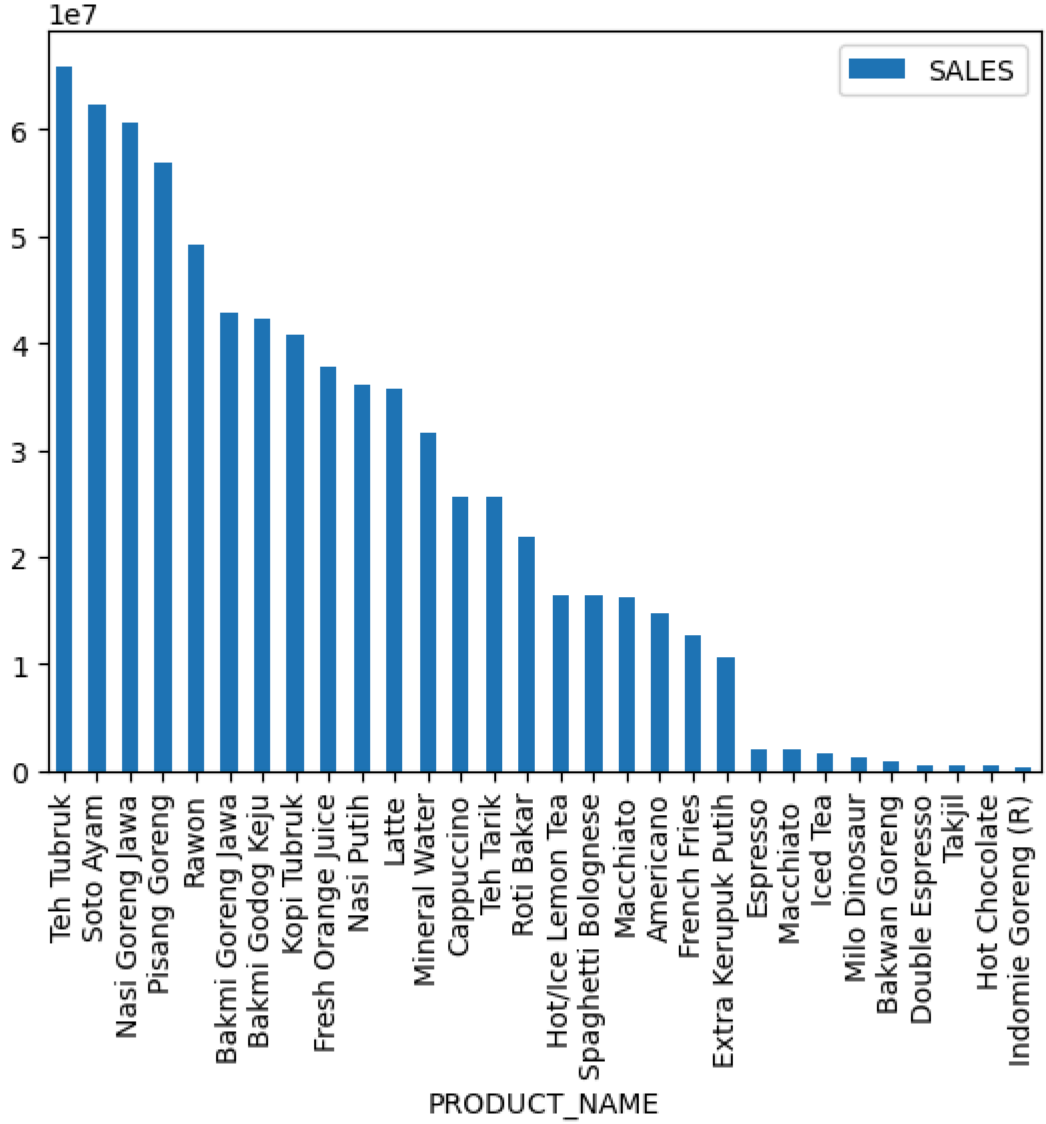
Out[ ]:

PRODUCT_NAME	SALES
Americano	14673000
Bakmi Godog Keju	42296000
Bakmi Goreng Jawa	42826000
Bakwan Goreng	832000
Cappuccino	25699000
Double Espresso	616000
Es Batu	10000
Es Teh Susu (Gula Jawa)	120000
Espresso	2030000
Extra Kerupuk Putih	10626000
French Fries	12667500
Fresh Orange Juice	37721000
Green Tea	300000
Hot Chocolate	450000
Hot/Ice Lemon Tea	16454000
Iced Peach Tea	100000
Iced Tea	1740000
Indomie Goreng (J)	364000
Indomie Goreng (R)	432000
Indomie Goreng Aceh	96000
Jasmine	100000
Kopi Tubruk	40846000
Latte	35753000
Lemon Tea	125000
Macchiato	16227000
Macchiato	2000000
Milo Dinosaur	1305000
Mineral Water	31554000
Nasi Goreng Jawa	60616000
Nasi Putih	36035378
Pisang Goreng	56886000
Pisang Goreng Tuna Sandwich	315000
Rawon	49199000
Roti Bakar	21926078
Soto Ayam	62339800
Spaghetti Bolognese	16400000

	SALES
PRODUCT_NAME	
Takjil	515000
Teh Tarik	25618000
Teh Tubruk	65868446
Telur Orak Arik	162000

In [ ]:
```
df1.sort_values(by=['SALES'],ascending=False).head(30).plot(kind='bar')
```

Out[ ]:
```
<Axes: xlabel='PRODUCT_NAME'>
```



Transformasi Data

In [ ]:
```
dfHotEncoded = df.pivot_table(index='TRX_ID', columns='PRODUCT_NAME', values
dfHotEncoded
```

Out[ ]:

PRODUCT_NAME	Americano	Bakmi Godog Keju	Bakmi Goreng Jawa	Bakwan Goreng	Cappuccino	Double Espresso	Es Batu	Es Teh Susu (Gula Jawa)
TRX_ID								
85	NaN	NaN	NaN	NaN	34000.0	NaN	NaN	NaN
86	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
87	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
88	NaN	NaN	NaN	NaN	34000.0	NaN	NaN	NaN
89	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
8848	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8849	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8850	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8851	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8852	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

8144 rows × 40 columns

In [ ]:
```
dfHotEncoded.shape
```

Out[ ]:
```
(8144, 40)
```

In [ ]:
```
dfHotEncoded = df.pivot_table(index='TRX_ID', columns='PRODUCT_NAME', values
dfHotEncoded
```

Out[ ]:

PRODUCT_NAME	Americano	Bakmi Godog Keju	Bakmi Goreng Jawa	Bakwan Goreng	Cappuccino	Double Espresso	Es Batu	Es Teh Susu (Gula Jawa)
TRX_ID								
85	0.0	0.0	0.0	0.0	34000.0	0.0	0.0	0.0
86	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
87	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
88	0.0	0.0	0.0	0.0	34000.0	0.0	0.0	0.0
89	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
8848	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8849	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8850	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8851	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

8144 rows × 40 columns

In [ ]:

```
dfHotEncoded[dfHotEncoded>0] = 1
dfHotEncoded
```

Out[ ]:

PRODUCT_NAME	Americano	Bakmi Godog Keju	Bakmi Goreng Jawa	Bakwan Goreng	Cappuccino	Double Espresso	Es Batu	Es Teh Susu (Gula Jawa)
TRX_ID								
85	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
86	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
87	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
88	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
89	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
8848	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8849	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8850	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8851	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

8144 rows × 40 columns

Menentukan Association Rule

Pada association rule, dua konsep pengukuran yang sangat penting untuk diperhatikan adalah support dan confidence.

Support menentukan seberapa sering suatu itemset muncul dalam data, diukur dalam skala nol hingga satu. Nol artinya tidak pernah muncul dalam data, sementara angka satu berarti itemset tersebut selalu muncul.

$$Support\ (a \rightarrow b) = \frac{jumlah\ transaksi\ yang\ melibatkan\ (a \rightarrow b)}{jumlah\ total\ transaksi}$$

"jumlah transaksi yang melibatkan (a -> b)" dihitung dari frekuensi kemunculan rule {a} -> {b} dalam dataset.

Confidence digunakan untuk menentukan akurasi suatu rule, atau bisa juga dianggap sebagai ukuran seberapa yakin (confident) kita terhadap suatu rule.

$$Confidence\ \ (a \rightarrow b) = \frac{Support\ (a \rightarrow b)}{Support\ (a)}$$

Apriori

Memilih frequent itemset

In [ ]:
```
df2 = apriori(dfHotEncoded, min_support =0.1, use_colnames=True)
df2
```

```
/Users/dafinazwa/anaconda3/lib/python3.10/site-packages/mlxtend/frequent_pa
tterns/fpcommon.py:110: DeprecationWarning: DataFrames with non-bool types
result in worse computationalperformance and their support might be discont
inued in the future.Please use a DataFrame with bool type
  warnings.warn(
```

Out[ ]:

	support	itemsets
0	0.131508	(Bakmi Godog Keju)
1	0.111370	(Bakmi Goreng Jawa)
2	0.129912	(Extra Kerupuk Putih)
3	0.111616	(Fresh Orange Juice)
4	0.177554	(Kopi Tubruk)
5	0.109283	(Latte)
6	0.224705	(Mineral Water)
7	0.170309	(Nasi Goreng Jawa)
8	0.196832	(Nasi Putih)
9	0.206655	(Pisang Goreng)
10	0.154101	(Rawon)
11	0.173502	(Soto Ayam)
12	0.342829	(Teh Tubruk)
13	0.104003	(Nasi Goreng Jawa, Mineral Water)
14	0.101793	(Teh Tubruk, Mineral Water)
15	0.122667	(Teh Tubruk, Pisang Goreng)

Memilih rule yang kuat

Angka support dan confidence lebih dari atau sama dengan min_support dan min_confidence.

In [ ]:

```
df3 = association_rules(df2, metric ='confidence', min_threshold=0.1)
df3
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(Nasi Goreng Jawa)	(Mineral Water)	0.170309	0.224705	0.104003	0.610671	2.717651	0.06
1	(Mineral Water)	(Nasi Goreng Jawa)	0.224705	0.170309	0.104003	0.462842	2.717651	0.06
2	(Teh Tubruk)	(Mineral Water)	0.342829	0.224705	0.101793	0.296920	1.321374	0.02
3	(Mineral Water)	(Teh Tubruk)	0.224705	0.342829	0.101793	0.453005	1.321374	0.02
4	(Teh Tubruk)	(Pisang Goreng)	0.342829	0.206655	0.122667	0.357808	1.731425	0.05
5	(Pisang Goreng)	(Teh Tubruk)	0.206655	0.342829	0.122667	0.593583	1.731425	0.05

In [ ]:

```
df3.sort_values(by=['confidence'], ascending=False)
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(Nasi Goreng Jawa)	(Mineral Water)	0.170309	0.224705	0.104003	0.610671	2.717651	0.06
5	(Pisang Goreng)	(Teh Tubruk)	0.206655	0.342829	0.122667	0.593583	1.731425	0.05
1	(Mineral Water)	(Nasi Goreng Jawa)	0.224705	0.170309	0.104003	0.462842	2.717651	0.06
3	(Mineral Water)	(Teh Tubruk)	0.224705	0.342829	0.101793	0.453005	1.321374	0.02
4	(Teh Tubruk)	(Pisang Goreng)	0.342829	0.206655	0.122667	0.357808	1.731425	0.05
2	(Teh Tubruk)	(Mineral Water)	0.342829	0.224705	0.101793	0.296920	1.321374	0.02

Mencari petunjuk hubungan dua itemset ketika muncul bersamaan dengan Lift

Lift digunakan untuk mengetahui apakah dua itemset muncul secara bersamaan karena suatu kebetulan atau tidak.

1. Lift = 1 artinya tidak ada hubungan apa-apa antara kedua itemset
2. Lift > 1 artinya ada hubungan positif antara kedua itemset dan keduanya muncul bukan kebetulan belaka
3. Lift < 1 artinya hubungan kedua itemset terlalu lemah.

In [ ]:
```
df4 = association_rules(df2, metric ='lift', min_threshold=1)
df4
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(Nasi Goreng Jawa)	(Mineral Water)	0.170309	0.224705	0.104003	0.610671	2.717651	0.06
1	(Mineral Water)	(Nasi Goreng Jawa)	0.224705	0.170309	0.104003	0.462842	2.717651	0.06
2	(Teh Tubruk)	(Mineral Water)	0.342829	0.224705	0.101793	0.296920	1.321374	0.02
3	(Mineral Water)	(Teh Tubruk)	0.224705	0.342829	0.101793	0.453005	1.321374	0.02
4	(Teh Tubruk)	(Pisang Goreng)	0.342829	0.206655	0.122667	0.357808	1.731425	0.05
5	(Pisang Goreng)	(Teh Tubruk)	0.206655	0.342829	0.122667	0.593583	1.731425	0.05

In [ ]:
```
df4.sort_values(by=['support'], ascending=False)
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
4	(Teh Tubruk)	(Pisang Goreng)	0.342829	0.206655	0.122667	0.357808	1.731425	0.05
5	(Pisang Goreng)	(Teh Tubruk)	0.206655	0.342829	0.122667	0.593583	1.731425	0.05
0	(Nasi Goreng Jawa)	(Mineral Water)	0.170309	0.224705	0.104003	0.610671	2.717651	0.06
1	(Mineral Water)	(Nasi Goreng Jawa)	0.224705	0.170309	0.104003	0.462842	2.717651	0.06
2	(Teh Tubruk)	(Mineral Water)	0.342829	0.224705	0.101793	0.296920	1.321374	0.02
3	(Mineral Water)	(Teh Tubruk)	0.224705	0.342829	0.101793	0.453005	1.321374	0.02

FP-Growth

In [ ]:

```
fp_growth =fpgrowth(dfHotEncoded, min_support=0.1, use_colnames=True)
fp_growth
```

```
/Users/dafinazwa/anaconda3/lib/python3.10/site-packages/mlxtend/frequent_pa
tterns/fpcommon.py:110: DeprecationWarning: DataFrames with non-bool types
result in worse computationalperformance and their support might be discont
inued in the future.Please use a DataFrame with bool type
  warnings.warn(
```

Out[ ]:

	support	itemsets
0	0.196832	(Nasi Putih)
1	0.177554	(Kopi Tubruk)
2	0.173502	(Soto Ayam)
3	0.154101	(Rawon)
4	0.224705	(Mineral Water)
5	0.111616	(Fresh Orange Juice)
6	0.170309	(Nasi Goreng Jawa)
7	0.129912	(Extra Kerupuk Putih)
8	0.109283	(Latte)
9	0.342829	(Teh Tubruk)
10	0.131508	(Bakmi Godog Keju)
11	0.111370	(Bakmi Goreng Jawa)
12	0.206655	(Pisang Goreng)
13	0.101793	(Teh Tubruk, Mineral Water)
14	0.104003	(Nasi Goreng Jawa, Mineral Water)
15	0.122667	(Teh Tubruk, Pisang Goreng)

In [ ]:
```
rules_fp = association_rules(fp_growth, metric="confidence", min_threshold=(
rules_fp
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
0	(Teh Tubruk)	(Mineral Water)	0.342829	0.224705	0.101793	0.296920	1.321374	0.02
1	(Mineral Water)	(Teh Tubruk)	0.224705	0.342829	0.101793	0.453005	1.321374	0.02
2	(Nasi Goreng Jawa)	(Mineral Water)	0.170309	0.224705	0.104003	0.610671	2.717651	0.06
3	(Mineral Water)	(Nasi Goreng Jawa)	0.224705	0.170309	0.104003	0.462842	2.717651	0.06
4	(Teh Tubruk)	(Pisang Goreng)	0.342829	0.206655	0.122667	0.357808	1.731425	0.05
5	(Pisang Goreng)	(Teh Tubruk)	0.206655	0.342829	0.122667	0.593583	1.731425	0.05

In [ ]:
```
rules_fp.sort_values(by=['support','confidence','lift'], ascending=False)
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
5	(Pisang Goreng)	(Teh Tubruk)	0.206655	0.342829	0.122667	0.593583	1.731425	0.05
4	(Teh Tubruk)	(Pisang Goreng)	0.342829	0.206655	0.122667	0.357808	1.731425	0.05
2	(Nasi Goreng Jawa)	(Mineral Water)	0.170309	0.224705	0.104003	0.610671	2.717651	0.06
3	(Mineral Water)	(Nasi Goreng Jawa)	0.224705	0.170309	0.104003	0.462842	2.717651	0.06
1	(Mineral Water)	(Teh Tubruk)	0.224705	0.342829	0.101793	0.453005	1.321374	0.02
0	(Teh Tubruk)	(Mineral Water)	0.342829	0.224705	0.101793	0.296920	1.321374	0.02