

**Kampus  
Merdeka**  
INDONESIA JAYA

**data  
acaemy**  
CYBERTREND DATA ACADEMY

2023

# Rekomendasi Produk Bundling dengan Metode Asosiasi untuk Optimalisasi Penjualan

Seaborn Skipy



DATA ACADEMY™



## Member Of Team



**Reggina Indriyani**

Sekolah Tinggi Ilmu  
Manajemen dan Ilmu  
Komputer ESQ



**Putri Nurul Fadilah**

Institut Teknologi Telkom  
Purwokerto



**Ahmad Kamil Jamal**

Universitas Indonesia



**Amalia Mayzahaq**

Universitas Pendidikan  
Indonesia



**DATA ACADEMY™**

**Kampus  
Merdeka**  
INDONESIA JAYA

**data  
acaemy**  
CYBERTREND DATA ACADEMY

# Business Understanding



DATA ACADEMY™



## Business Understanding

Objective Problem : Menentukan produk bundling dan Penyusunan rak berdasarkan produk yang memiliki hubungan.

- Problem Data Science : Association Rules.
- Algoritma : Apriori dan FP-Growth.

**Kampus  
Merdeka**  
INDONESIA JAYA

**data  
acaemy**  
CYBERTREND DATA ACADEMY

# Data Understanding



DATA ACADEMY™

# Data Understanding

Transaction	Item		date_time	period_day	weekday_weekend
0	1	Bread	30-10-2016 09:58	morning	weekend
1	2	Scandinavian	30-10-2016 10:05	morning	weekend
2	2	Scandinavian	30-10-2016 10:05	morning	weekend
3	3	Hot chocolate	30-10-2016 10:07	morning	weekend
4	3	Jam	30-10-2016 10:07	morning	weekend
...	...	...	...	...	...
20502	9682	Coffee	09-04-2017 14:32	afternoon	weekend
20503	9682	Tea	09-04-2017 14:32	afternoon	weekend
20504	9683	Coffee	09-04-2017 14:57	afternoon	weekend
20505	9683	Pastry	09-04-2017 14:57	afternoon	weekend
20506	9684	Smoothies	09-04-2017 15:04	afternoon	weekend

20507 rows × 5 columns

Dataset ini milik sebuah toko roti dengan nama “The Bread Basket”. Dataset memiliki 20507 baris dan 5 kolom.

Dataset ini periode waktu datanya adalah **30-10-2016 hingga 09-04-2017**.

Sumber Data : Kaggle

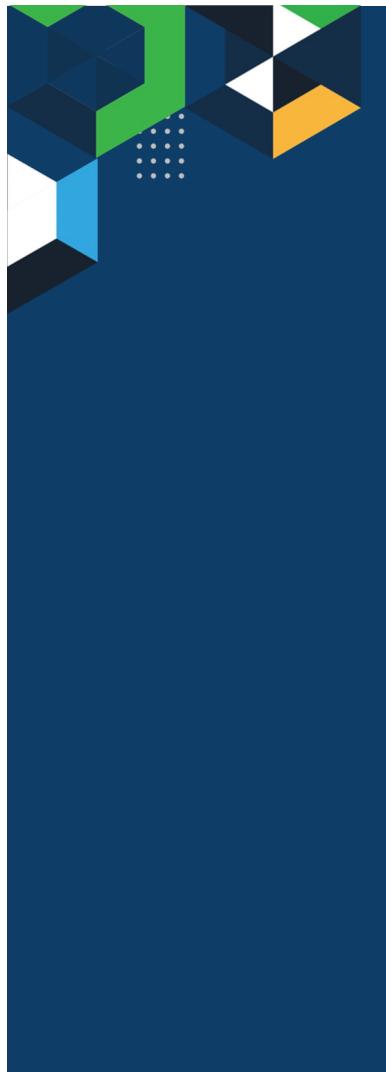
**Kampus  
Merdeka**  
INDONESIA JAYA

**data  
acaemy**  
CYBERTREND DATA ACADEMY

# Data Preparation



DATA ACADEMY™



```
[ ] #Cek missing value
print(df.isnull().sum().sort_values(ascending = False))

Transaction      0
Item            0
date_time       0
period_day      0
weekday_weekend 0
dtype: int64

Tidak ada nilai null dalam setiap kolom DataFrame

[ ] #Mengonversi kolom 'date_time' menjadi tipe data datetime
df['date_time'] = pd.to_datetime(df['date_time'], format='%d-%m-%Y %H:%M')

#Membuat kolom 'Date' yang hanya berisi tanggal
df['Date'] = df['date_time'].dt.date

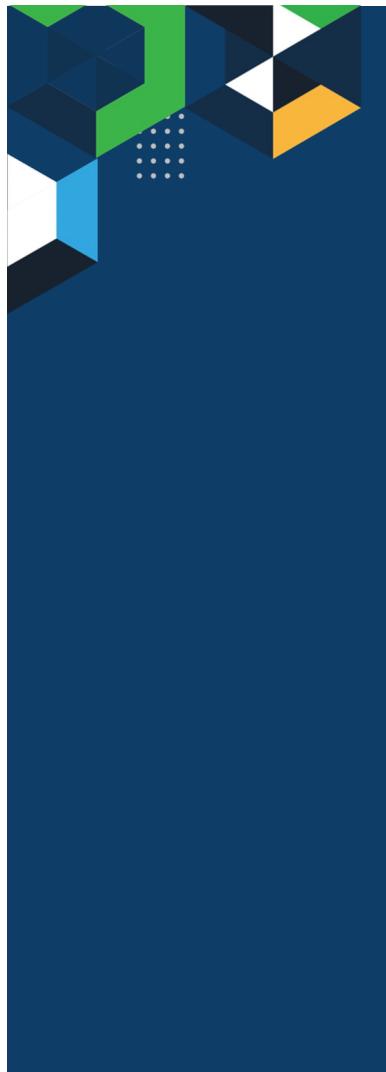
#Membuat kolom 'Time' yang hanya berisi waktu
df['Time'] = df['date_time'].dt.time

#Menampilkan DataFrame hasil
print(df[['Date', 'Time']])
```

```
df['month'] = df['date_time'].dt.month
df['month name'] = df['month'].replace([1,2,3,4,5,6,7,8,9,10,11,12],['January',
                                         'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'])

df['Date'] = pd.to_datetime(df['Date'])
#Extract year, month, day ke dalam kolom baru
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
```

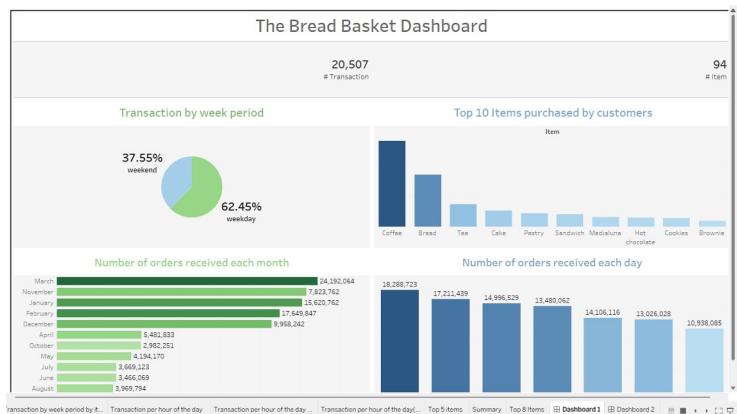
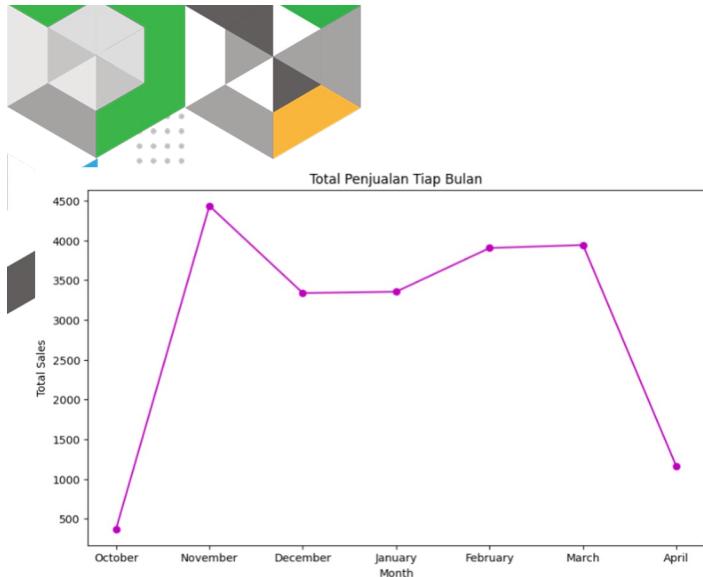
- Dataset tidak memiliki missing value.
- Memisahkan antara kolom Date & Time.
- Melakukan feature extraction (Date, time, hour, month, month name, Year, day, weekday, weekday name).



Melakukan normalisasi dan Memastikan data (item) bersih dari whitespace agar memudahkan untuk dianalisis.

```
#Menerapkan pandas apply
df["Item"] = df["Item"].apply(lambda item: item.lower())
```

```
#Memastikan setiap nilai "Item" bersih dari whitespace
df["Item"] = df["Item"].apply(lambda item: item.strip())
```



# Analisis Sederhana

Melakukan analisis sederhana untuk :

- Memahami data dengan lebih baik.
- Mengidentifikasi pola dan tren dalam data.
- Mengidentifikasi masalah data.
- Memilih model yang tepat.

**Kampus  
Merdeka**  
INDONESIA JAYA

**data  
acaemy**  
CYBERTREND DATA ACADEMY

# Data Modeling



DATA ACADEMY™



# Apriori

Algoritma Apriori adalah salah satu algoritma asosiasi untuk menemukan aturan asosiasi yang menggambarkan hubungan antara item dalam data. Berikut tahapannya :

## Membuat list transaksi ('Transaction' dan 'Item')

```
#Menggunakan loop untuk membuat list transaksi dalam dataset
transaction_list = []

for i in df['Transaction']:
    tlist = list(set(df[df['Transaction'] == i]['Item']))
    if len(tlist) > 0:
        transaction_list.append(tlist)
```

## Menerapkan transaction encoder

```
from mlxtend.preprocessing import TransactionEncoder
te = TransactionEncoder()
te_ary = te.fit(transaction_list).transform(transaction_list)
df2 = pd.DataFrame(te_ary, columns=te.columns_)
```

## Membuat model apriori

```
#Library Apriori
from mlxtend.frequent_patterns import association_rules, apriori

#Menerapkan apriori
apriori = apriori(df2, min_support=0.01, use_colnames=True)

#Aturan asosiasi dengan metriks lift
rules = association_rules(apriori, metric='lift', min_threshold=1.0)

metrix = pd.DataFrame(rules)
result = metrix[['antecedents', 'consequents', 'support','confidence',
                 'lift', 'leverage', 'conviction']].sort_values('lift', ascending=False)

#Menyimpan hasil ke file CSV
result.to_csv('Aturan_Apriori.csv', index=False)

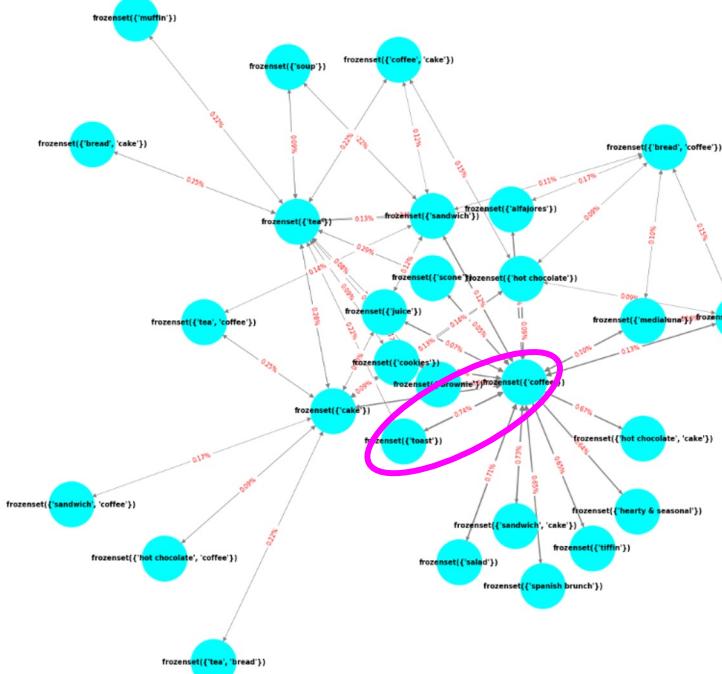
#Styling Output Agar Rapih
styled_result = result.style.format({
    'confidence': '{:.2%}',
    'lift': '{:.2f}',
    'leverage': '{:.2f}',
    'conviction': '{:.2f}',
})
display (styled_result)
```

## Hasil

antecedents	consequents	support	confidence	lift	leverage	conviction
frozenset({'juice'})	frozenset({'cookies'})	0.013898	19.99%	2.41	0.01	1.15
frozenset({'cookies'})	frozenset({'juice'})	0.013898	16.75%	2.41	0.01	1.12
frozenset({'sandwich'})	frozenset({'soup'})	0.011021	10.10%	1.98	0.01	1.06
frozenset({'soup'})	frozenset({'sandwich'})	0.011021	21.59%	1.98	0.01	1.14
frozenset({'juice'})	frozenset({'sandwich'})	0.013313	19.14%	1.75	0.01	1.10
frozenset({'sandwich'})	frozenset({'juice'})	0.013313	12.20%	1.75	0.01	1.06
frozenset({'hot chocolate'})	frozenset({'cookies'})	0.012874	14.50%	1.75	0.01	1.07
frozenset({'cookies'})	frozenset({'hot chocolate'})	0.012874	15.52%	1.75	0.01	1.08
frozenset({'soup'})	frozenset({'tea'})	0.016531	32.38%	1.69	0.01	1.19
frozenset({'tea'})	frozenset({'soup'})	0.016531	8.61%	1.69	0.01	1.04

Apriori menghasilkan 102 aturan

# Membuat Network Diagram



```
#Buat graph dari aturan asosiasi
G = nx.DiGraph()
for i, row in rules.iterrows():
    G.add_edge(str(row['antecedents']), str(row['consequents']), weight=row['confidence'])

#Posisi node
pos = nx.spring_layout(G, k=0.2)
plt.figure(figsize=(20, 14))
nx.draw_networkx(G, pos, with_labels=True, node_size=2000, node_color='cyan',
                  font_size=7, font_color='black', font_weight='bold', edge_color='gray',
                  width=[d['weight']*2 for u, v, d in G.edges(data=True)])

#Label edge(persentasi nilai confidence)
edge_labels = {(u, v): f"{d['weight']:.2f}%" for u, v, d in G.edges(data=True)}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='red', font_size=6)
```

Kita mengambil contoh dengan nilai persentase tertinggi, yang dimiliki oleh node "Coffee" yang memiliki panah bolak-balik ke node "toast", menunjukkan hubungan yang bersifat saling ketergantungan antar keduanya. Artinya, konsumsi "Coffee" mempengaruhi "toast", dan sebaliknya. Angka confidence 0,74% menunjukkan bahwa aturan asosiasi antara "Coffee" dan "toast" terbukti benar sekitar 0,74% dari semua transaksi.



# FP-Growth

FP-Growth (Frequent Pattern Growth) adalah algoritma untuk menemukan itemset yang sering muncul atau pola-pola yang sering terjadi dalam kumpulan data transaksional. Fp-growth dirancang untuk mengatasi keterbatasan asosiasi tradisional (Apriori), terutama dalam hal kinerja.

## Menerapkan transaction encoder ('Transaction' dan 'Item')

```
te = TransactionEncoder()
te_ary = te.fit(df.groupby('Transaction')['Item'].apply(list)).transform(df.groupby('Transaction')['Item'].apply(list))
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)
```

## Menerapkan FP-Growth

```
#Menerapkan FP-Growth pada DataFrame yang sudah diencode
fpgrowth_result = fpgrowth(df_encoded, min_support=0.01, use_colnames=True)

# Menghasilkan aturan asosiasi dengan metriks lift
rules_fp = association_rules(fpgrowth_result, metric='lift', min_threshold=1.0)
rules_fp[ 'support' ] = rules_fp[ 'support' ]
rules_fp[ 'leverage' ] = rules_fp[ 'leverage' ]
rules_fp[ 'conviction' ] = rules_fp[ 'conviction' ]
```

## Menampilkan aturan FP-Growth

```
#Menampilkan hasil diurutkan berdasarkan nilai lift
hasil = rules_fp[['antecedents', 'consequents', 'support', 'confidence', 'lift',
                  'leverage', 'conviction']].sort_values('lift', ascending=False)

#Menyimpan hasil ke file CSV
hasil.to_csv('Aturan_FPGrowth.csv', index=False)

#Styling Output Agar Rapih
result = result.style.format({
    'confidence': '{:.2%}',
    'lift': '{:.2f}',
    'leverage': '{:.2f}',
    'conviction': '{:.2f}',
})
display (result)
```

## Hasil

antecedents	consequents	support	confidence	lift	leverage	conviction
frozenset({'juice'})	frozenset({'cookies'})	0.013898	19.99%	2.41	0.01	1.15
frozenset({'cookies'})	frozenset({'juice'})	0.013898	16.75%	2.41	0.01	1.12
frozenset({'sandwich'})	frozenset({'soup'})	0.011021	10.10%	1.98	0.01	1.06
frozenset({'soup'})	frozenset({'sandwich'})	0.011021	21.59%	1.98	0.01	1.14
frozenset({'juice'})	frozenset({'sandwich'})	0.013313	19.14%	1.75	0.01	1.10
frozenset({'sandwich'})	frozenset({'juice'})	0.013313	12.20%	1.75	0.01	1.06
frozenset({'hot chocolate'})	frozenset({'cookies'})	0.012874	14.50%	1.75	0.01	1.07
frozenset({'cookies'})	frozenset({'hot chocolate'})	0.012874	15.52%	1.75	0.01	1.08
frozenset({'soup'})	frozenset({'tea'})	0.016531	32.38%	1.69	0.01	1.19
frozenset({'tea'})	frozenset({'soup'})	0.016531	8.61%	1.69	0.01	1.04

FP-Growth menghasilkan 42 aturan

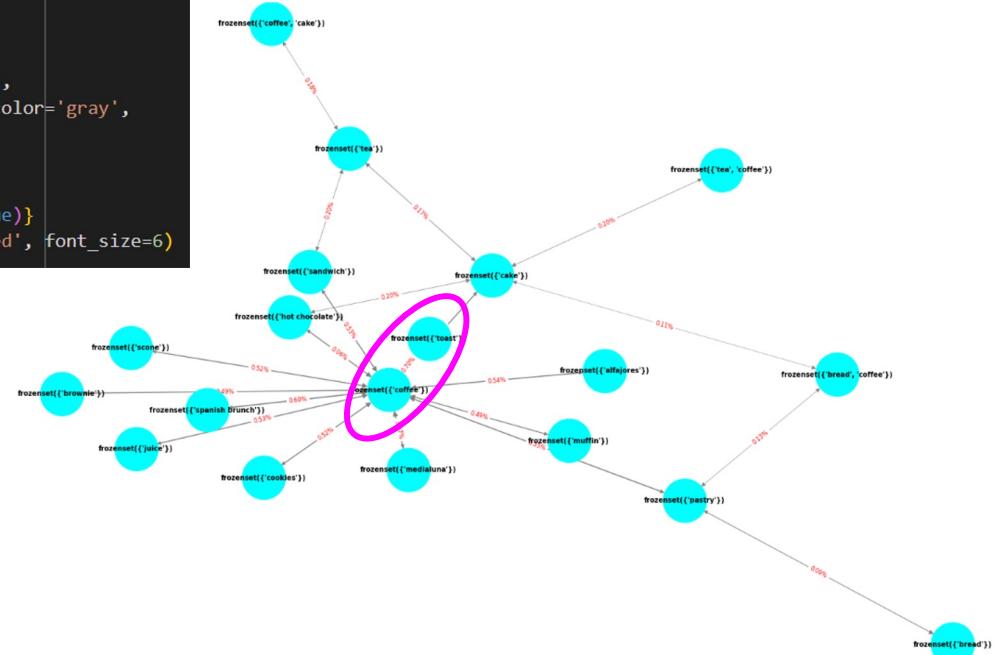
## Membuat Network Diagram

```
#Buat graph dari aturan asosiasi
G = nx.DiGraph()
for i, row in rules.iterrows():
    G.add_edge(str(row['antecedents']), str(row['consequents']), weight=row['confidence'])

#Atur posisi node
pos = nx.spring_layout(G, k=0.2)
plt.figure(figsize=(20, 14))
nx.draw_networkx(G, pos, with_labels=True, node_size=2000, node_color='cyan',
                 font_size=7, font_color='black', font_weight='bold', edge_color='gray',
                 width=[d['weight']*2 for u, v, d in G.edges(data=True)])

#Label edge (Persentasi nilai confidence)
edge_labels = {(u, v): f"{d['weight']:.2f}%" for u, v, d in G.edges(data=True)}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='red', font_size=6)
```

Persentase tertinggi dimiliki oleh node "Coffee" yang memiliki panah bolak-balik ke node "toast". Konsumsi "Coffee" mempengaruhi "toast", dan sebaliknya. Angka confidence 0,70% menunjukkan bahwa aturan asosiasi antara "Coffee" dan "toast" terbukti benar sekitar 0,70% dari semua transaksi.



**Kampus  
Merdeka**  
INDONESIA JAYA

**data  
acaemy**  
CYBERTREND DATA ACADEMY

# Data Visualization



DATA ACADEMY™

unsaved filter(s)



## Rekomendasi Produk Bundling

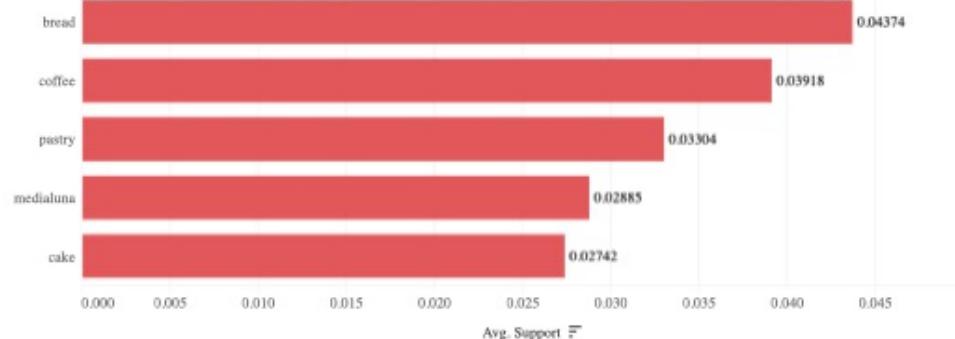
With Apriori

### Association Rules

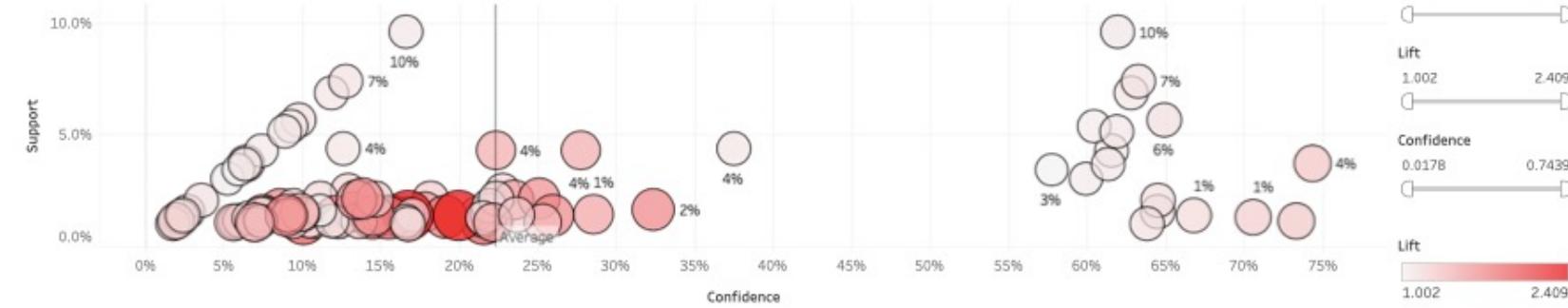
Antecedents	Consequents	Confidence	Support	Lift
alfajores	coffee	61.35%	3.68%	1.06
	coffee, bread	16.76%	1.00%	1.16
	tea	23.68%	1.42%	1.23
bread	pastry	12.63%	4.37%	1.08
	cake	22.18%	1.08%	1.43
brownie	coffee	57.76%	3.43%	1.00
	tea	21.53%	1.28%	1.12
cake	bread, tea	7.00%	1.08%	1.43
	coffee	61.96%	9.59%	1.07
	coffee, hot chocolate	9.01%	1.39%	1.68
	coffee, sandwich	7.37%	1.14%	1.08
	coffee, tea	13.87%	2.15%	1.62
	cookies	9.14%	1.41%	1.10
	hot chocolate	12.49%	2.00%	1.52

Product (All)

### Product Associate to All



### Association Rules by Confidence, Support, and Lift



a b | e u

a<sup>0</sup>



## Rekomendasi Produk Bundling

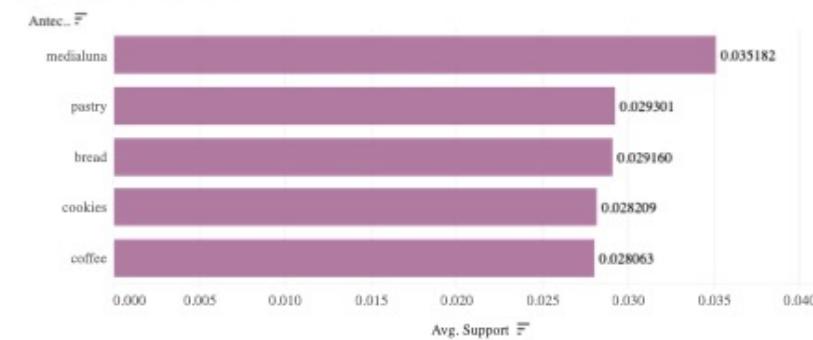
With FP-Growth

### Association Rules

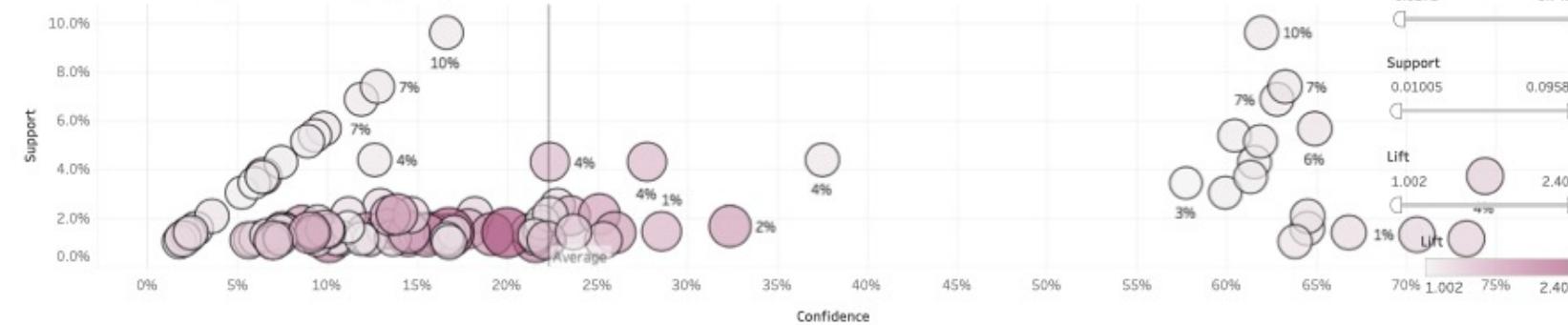
Antecedents	Consequents	Confiden..	Support	Lift	
alfajores	coffee	54.07%	1.97%	1.13	Abc
bread	pastry	8.91%	2.92%	1.03	Abc
brownie	coffee	49.08%	1.97%	1.03	Abc
cake	coffee	52.70%	5.47%	1.10	Abc
	tea	22.89%	2.38%	1.60	Abc
	hot chocolate	10.99%	1.14%	1.88	Abc
	coffee, tea	9.66%	1.00%	1.94	Abc
	coffee, bread	9.66%	1.00%	1.07	Abc
coffee	cake	11.44%	5.47%	1.10	Abc
	pastry	9.94%	4.75%	1.15	Abc
	sandwich	7.99%	3.82%	1.11	Abc
	medialuna	7.35%	3.52%	1.19	Abc
	hot chocolate	6.18%	2.96%	1.06	Abc

Product (All)

### Product Associate to All



### Association Rules by Confidence, Support, and Lift



⋮ + a b | e a u

⋮ ↴



## Association Rules

Antecedents	Consequents	Confid..	Support	Lift
alfajores	coffee	61,35%	3,68%	1,06
	coffee, bread	16,76%	1,00%	1,16
	tea	23,68%	1,42%	1,23
bread	pastry	12,63%	4,37%	1,08
	cake	22,18%	1,08%	1,43
brownie	coffee	57,76%	3,43%	1,00
	tea	21,53%	1,28%	1,12
cake	bread, tea	7,00%	1,08%	1,43
	coffee	61,96%	9,59%	1,07
	coffee, hot ch..	9,01%	1,39%	1,68
	coffee, sandw..	7,37%	1,14%	1,08
	coffee tea	13,87%	2,15%	1,62

DATA ACADEMY™

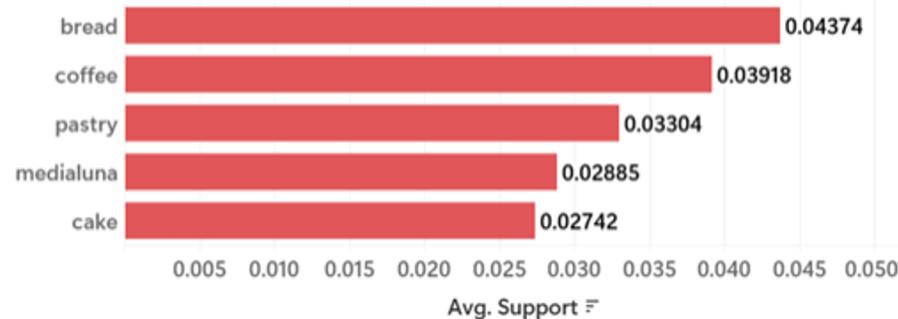
Tabel ini menunjukkan nilai confidence, support dan lift dari produk (Consequents) yang berasosiasi dengan produk (Antecedents)

Dengan keterangan

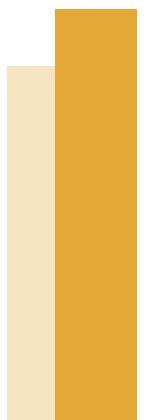
- Confidence: Probabilitas bahwa produk (Consequents) muncul dalam transaksi yang juga mengandung produk (Antecedents)
- Support: Proporsi transaksi dalam dataset yang mengandung produk tersebut
- Lift: Mengukur seberapa lebih sering produk (Consequents) muncul ketika produk (Antecedents) terjadi dibandingkan dengan frekuensi munculnya produk (Consequents)

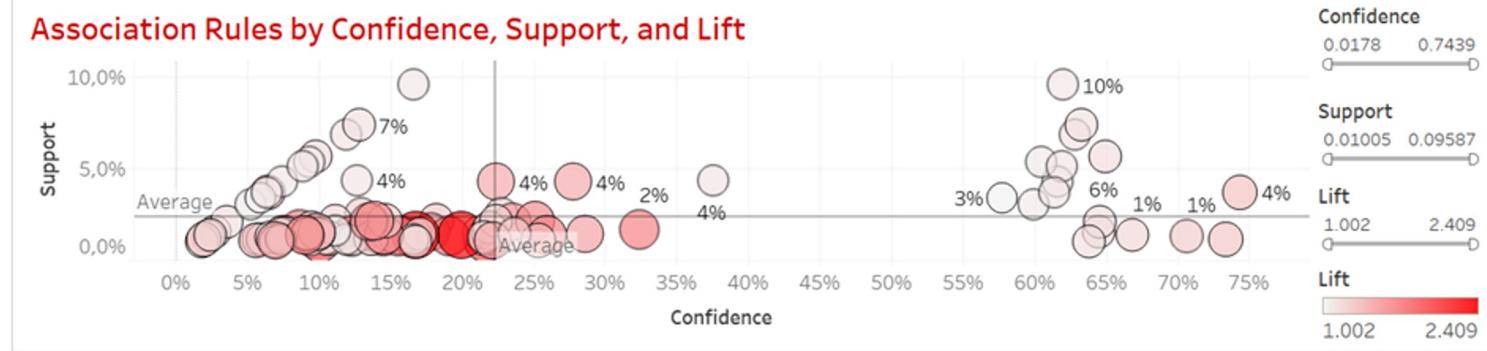


### Top 5 Product Associate to All



Grafik ini menunjukkan 5 produk (Consequents) teratas yang berasosiasi dengan produk (Attendance) berdasarkan nilai support.





Grafik ini menunjukkan persebaran nilai confidence, support dan lift.

# Contoh Pembacaan Dashboard

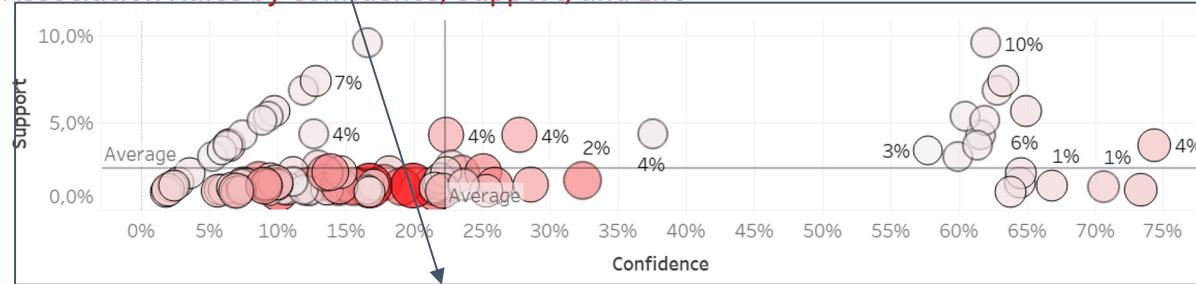
## Rekomendasi Produk Bundling

With Apriori

### Association Rules

Antecedents	Consequents	Confid..	Support	Lift
alfajores	coffee	61,35%	3,68%	1,06
brownie	coffee	57,76%	3,43%	1,00
cake	coffee	61,96%	9,59%	1,07
cake, hot chocol..	coffee	66,82%	1,39%	1,16
cake, sandwich	coffee	73,35%	1,14%	1,27
cookies	coffee	61,90%	5,13%	1,07
hearty & seasonal	coffee	63,83%	1,02%	1,11
hot chocolate	coffee	60,46%	5,37%	1,05
juice	coffee	1,57%	4,28%	1,07
medialuna	coffee	64,93%	5,65%	1,13
pastry	coffee	63,28%	7,38%	1,10
salad	coffee	70,60%	1,31%	1,22

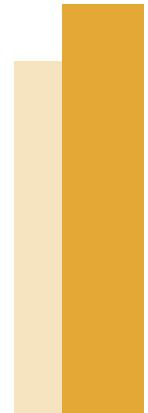
### Association Rules by Confidence, Support, and Lift



Jika membeli Alfajores maka kemungkinan 3,68% juga akan membeli Coffee dengan kepastian yang akan terjadi yaitu 61,35% berdasarkan transaksi yang sudah terjadi selama ini.

Produk (Consequents) yang ingin dibundling.

Cake merupakan produk yang paling baik untuk dibundling dengan Coffee berdasarkan nilai support.





## Kesimpulan

1. Dengan menerapkan aturan Asosiasi menggunakan metode Apriori dan FP Growth, kita berhasil mengidentifikasi pola-pola hubungan antarproduk berdasarkan transaksi yang terjadi pada rentang waktu 30 Oktober 2016 hingga 9 April 2017. Dari hasil analisis tersebut, dapat dilihat produk-produk yang memiliki keterkaitan yang signifikan.
2. Untuk performa model, algoritma FP Growth memiliki waktu eksekusi yang lebih cepat dibandingkan Apriori. Hal ini mengindikasikan efisiensi dalam pengolahan data, memberikan kelebihan dalam kinerja algoritma.
3. Aturan yang dihasilkan dari analisis ini dapat menjadi acuan bagi pemilik toko. Mereka dapat menggunakan informasi ini untuk merancang bundling produk yang lebih efektif, meningkatkan strategi penjualan, dan mendukung optimalisasi penjualan.



# Thank You



DATA ACADEMY™