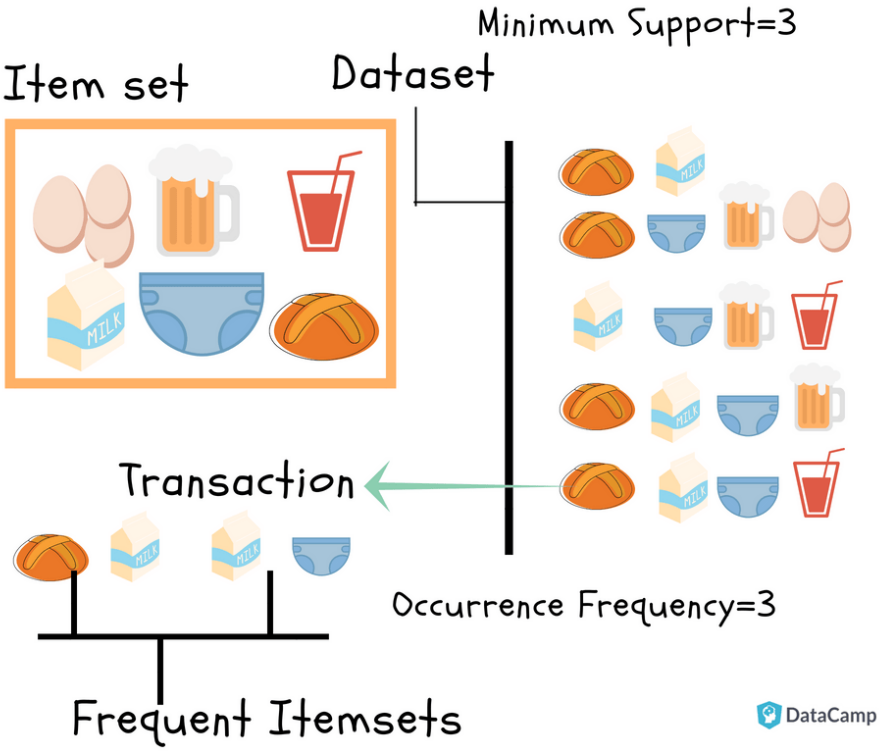
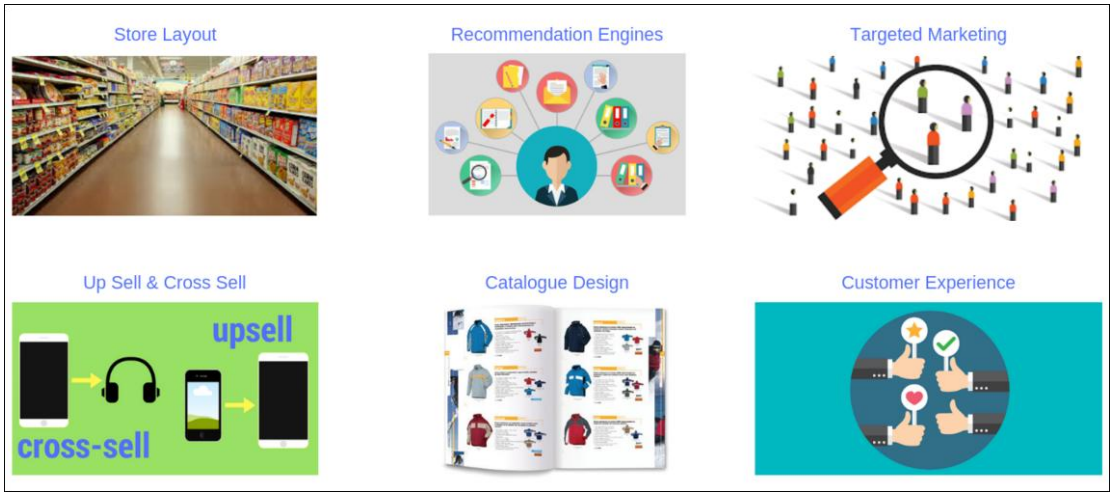
Nama Dosen : Teguh Iman Hermanto, M.Kom
Mata Kuliah : Machine Learning 1
Pembahasan : Algoritma Association Rules
Pokok Pemb : - Mengenal Market Basket analysis
- Membangun Model Algoritma Apriori
- Membangun Model Algoritma FP-Growth

1. Market Basket Analysis





2. Load dataset pada file Notebook

```python
import pandas as pd
import matplotlib.pyplot as plt
from mlxtend.frequent_patterns import apriori, fpgrowth
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
```

```python
# Load data
file_path = "Groceries_dataset.csv"
data = pd.read_csv(file_path)
```

3. Analisa Data

```python
# Tampilkan informasi awal dataset
print("Preview Dataset:")
print(data.head())
print("\n\nInfo Dataset:")
print(data.info())
print("\nStatistik Deskriptif Kolom Numerik:")
print(data.describe())
```

```python
# 1. Analisis kolom 'Member_number'
# Menghitung total member
total_member = data['Member_number'].nunique()
print(f"\n\nTotal member: {total_member}")
```

```python
# Top 5 anggota dengan transaksi terbanyak
top_members = data['Member_number'].value_counts().head()
print("\nTop 5 anggota dengan transaksi terbanyak:")
print(top_members)
```

```python
# 2. Analisis Kolom `Date`
data['Date'] = pd.to_datetime(data['Date'], format='%d-%m-%Y')

# Rentang waktu dataset
start_date = data['Date'].min()
end_date = data['Date'].max()
print(f"\nRentang waktu data: {start_date} hingga {end_date}")
```

```python
# Hitung jumlah transaksi per bulan
monthly_transactions = data['Date'].dt.to_period('M').value_counts().sort_index()
print("\nJumlah transaksi per bulan:")
print(monthly_transactions)
```

```python
# Visualisasi jumlah transaksi per bulan
plt.figure(figsize=(12, 6))
monthly_transactions.plot(kind='bar',
                          title='Transaksi per Bulan',
                          color='skyblue')
plt.xlabel('Bulan')
plt.ylabel('Jumlah Transaksi')
plt.xticks(rotation=45)
plt.show()
```

```python
# 3. Analisis Kolom `itemDescription`
# Barang paling sering dibeli
item_counts = data['itemDescription'].value_counts()
print("\nTop 10 barang paling sering dibeli:")
print(item_counts.head(10))
```

```python
# Visualisasi 10 barang teratas
plt.figure(figsize=(10, 6))
item_counts.head(10).plot(kind='bar',
                          title='Top 10 Barang Paling Sering Dibeli',
                          color='orange')
plt.xlabel('Barang')
plt.ylabel('Frekuensi')
plt.xticks(rotation=45)
plt.show()
```

```python
# 4. Analisis Gabungan
# Barang yang paling sering dibeli oleh anggota tertentu
member_id = 1808
member_items = data[data['Member_number'] == member_id]['itemDescription'].value_counts()
print(f"\nBarang paling sering dibeli oleh anggota {member_id}:")
print(member_items)
```

```python
# Pola pembelian barang pada bulan tertentu
month = '2015-07'
monthly_items = data[data['Date'].dt.to_period('M') == month]['itemDescription'].value_counts()
print(f"\nBarang paling sering dibeli pada {month}:")
print(monthly_items)
```

```python
# Statistik tambahan: jumlah barang unik
print("\nJumlah barang unik:")
print(data['itemDescription'].nunique())
```

4. Data Preprocessing

```
# Persiapan data: Mengelompokan item berdasarkan Member_number dan Date
transactions = data.groupby(['Member_number', 'Date'])['itemDescription'].apply(list)
```

```
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)
```

5. Modeling

```
# --- Algoritma Apriori ---
print("\n===== Apriori =====")
frequent_itemsets_apriori = apriori(df_encoded, min_support=0.005, use_colnames=True)
print("Frequent Itemsets (Apriori):")
print(frequent_itemsets_apriori)
```

```
# --- Algoritma FP-Growth ---
print("\n===== FP-Growth =====")
frequent_itemsets_fpgrowth = fpgrowth(df_encoded, min_support=0.005, use_colnames=True)
print("Frequent Itemsets (FP-Growth):")
print(frequent_itemsets_fpgrowth)
```

```
# Menampilkan hasil aturan asosiasi
print("\nAturan Asosiasi (Apriori):")
rules_apriori = association_rules(frequent_itemsets_apriori, metric="confidence", min_threshold=0.1)
print(rules_apriori[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```

```
print("\nAturan Asosiasi (FP-Growth):")
rules_fpgrowth = association_rules(frequent_itemsets_fpgrowth, metric="confidence", min_threshold=0.1)
print(rules_fpgrowth[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```