Nama : Sarwahita Dwi Prasanti
NIM : A11.2022.13987
Kelp : A11.4504

# Market Basket Analysis Apriori

- **Permasalahan**

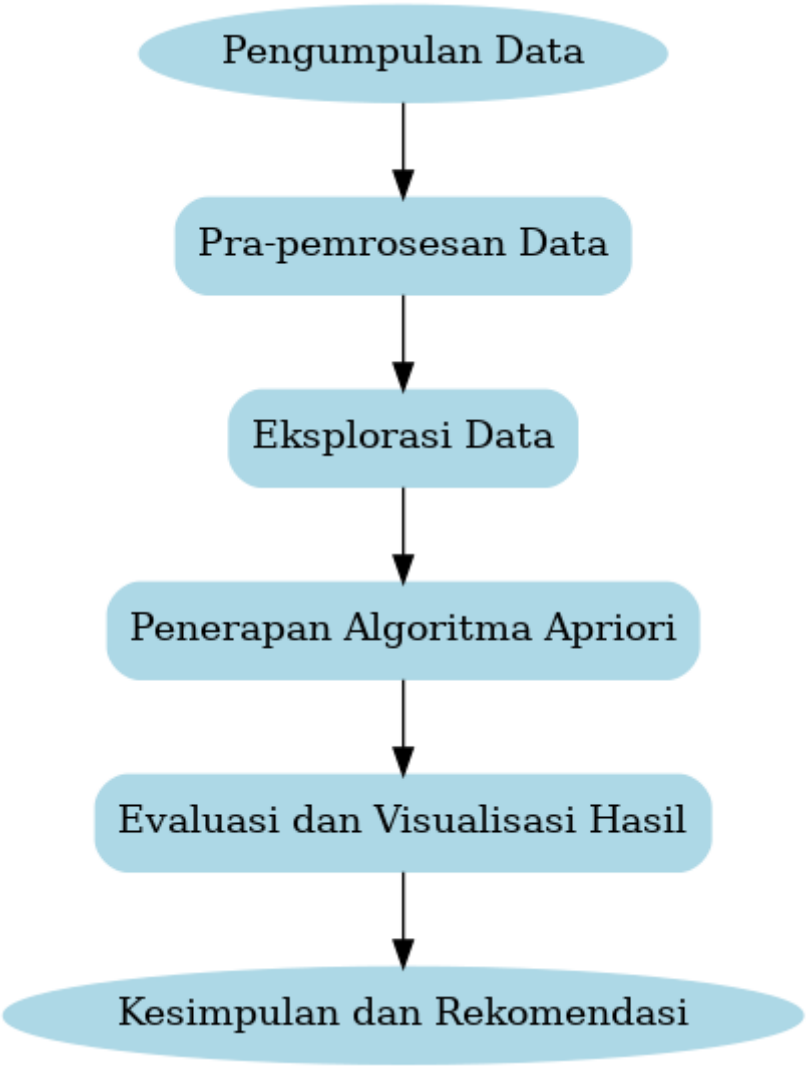
Dalam bisnis retail, memahami pola pembelian pelanggan sangat penting untuk meningkatkan strategi pemasaran, memaksimalkan penjualan, dan menyediakan rekomendasi produk. Namun, data transaksi yang besar dan beragam menyulitkan perusahaan untuk mengidentifikasi pola pembelian secara manual.

- **Tujuan**

1. Mengidentifikasi pola pembelian atau asosiasi produk yang sering muncul bersama dalam transaksi.
2. Memberikan rekomendasi produk kepada pelanggan berdasarkan data transaksi sebelumnya.
3. Meningkatkan strategi pemasaran melalui pengetahuan tentang pola pembelian pelanggan.

- **Alur Kerja**

- **Pengumpulan Data**: Data transaksi pelanggan yang mencakup kolom seperti ID Transaksi, Produk, Kuantitas, Tanggal, Harga, dan lainnya.
- **Pra-pemrosesan Data**:
  - Membersihkan data dari entri duplikat atau data yang tidak relevan.
  - Menstandarisasi format data untuk memastikan konsistensi.
  - Mengubah data transaksi ke format yang sesuai untuk analisis asosiasi (e.g., transaksi per produk).
- **Eksplorasi Data**:
  - Menganalisis distribusi transaksi, produk yang paling sering dibeli, dan tren pembelian.
- **Penerapan Algoritma Apriori**:
  - Menggunakan algoritma Apriori untuk menemukan aturan asosiasi antara produk.
  - Menentukan parameter minimum support dan confidence untuk analisis.
- **Evaluasi dan Visualisasi Hasil**:
  - aturan asosiasi yang dihasilkan.
  - Menampilkan pola pembelian melalui visualisasi seperti grafik atau tabel.
- **Kesimpulan dan Rekomendasi**:
  - Menginterpretasikan hasil untuk memberikan rekomendasi kepada manajemen.




- **Dataset**
  Dataset yang digunakan dalam analisis ini bersumber dari Kaggle, dengan nama **"Market Basket Analysis"** yang tersedia di tautan berikut: https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis/data.

  Dataset ini mencatat transaksi yang dilakukan oleh pelanggan di suatu toko. Setiap baris dalam dataset menggambarkan satu transaksi dan mencakup informasi seperti:

  1. **Transaction**: ID unik dari setiap transaksi.
  2. **Product**: Nama produk yang dibeli dalam transaksi tersebut.
  3. **Quantity**: Jumlah barang yang dibeli dalam satu transaksi.
  4. **Date** *(opsional)*: Tanggal transaksi dilakukan.
  5. **Price** *(opsional)*: Harga barang per unit.
  6. **CustomerID** *(opsional)*: ID pelanggan yang melakukan transaksi.
  7. **Country** *(opsional)*: Negara tempat transaksi dilakukan.

- **EDA**

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
import pickle
```

Library yang digunakan :

- pandas: Untuk manipulasi data seperti membaca file CSV dan analisis data tabular.
- numpy: Untuk operasi numerik.
- matplotlib.pyplot: Untuk membuat visualisasi data.
- mlxtend.preprocessing.TransactionEncoder: Untuk encoding transaksi dalam analisis Apriori.
- mlxtend.frequent_patterns: Untuk menjalankan algoritma Apriori dan menghasilkan aturan asosiasi.
- pickle: Untuk menyimpan atau memuat data.

[2]:
```
# 2. Baca Dataset
data = pd.read_csv('Assignment-1_Data.csv', delimiter=';', decimal=',', low_memory=False)
print("Data Awal:")
print(data.head())
```

```
Data Awal:
   BillNo                             Itemname  Quantity         Date  \
0  536365   WHITE HANGING HEART T-LIGHT HOLDER         6  01.12.2010 08:26
1  536365                  WHITE METAL LANTERN         6  01.12.2010 08:26
2  536365       CREAM CUPID HEARTS COAT HANGER         8  01.12.2010 08:26
3  536365  KNITTED UNION FLAG HOT WATER BOTTLE         6  01.12.2010 08:26
4  536365       RED WOOLLY HOTTIE WHITE HEART.         6  01.12.2010 08:26

   Price  CustomerID         Country
0   2.55     17850.0  United Kingdom
1   3.39     17850.0  United Kingdom
2   2.75     17850.0  United Kingdom
3   3.39     17850.0  United Kingdom
4   3.39     17850.0  United Kingdom
```

Membaca Dataset

Dataset yang digunakan memiliki beberapa kolom penting, seperti BillNo (ID transaksi), Itemname (nama produk), Quantity (jumlah produk yang dibeli), Date (tanggal transaksi), Price (harga per unit produk), CustomerID (ID pelanggan), dan Country (negara asal transaksi). Dataset dibaca dengan fungsi pd.read_csv, yang diatur dengan parameter khusus, seperti delimiter=';' untuk menyesuaikan pemisah data dan decimal=',' untuk mengubah format angka desimal yang menggunakan koma menjadi titik. Langkah ini dilanjutkan dengan menampilkan lima baris pertama data menggunakan head() untuk memverifikasi struktur dataset dan memastikan bahwa data telah dimuat dengan benar.

[3]:
```
# 3. Bersihkan Data
print("\nCek Missing Values:")
print(data.isnull().sum())
```

```
Cek Missing Values:
BillNo            0
Itemname       1455
Quantity          0
Date              0
Price             0
CustomerID    134041
Country           0
dtype: int64
```

Membersihkan Data

Setelah datanya kebaca, dicek lagi apakah ada nilai yang kosong (NaN) di tiap kolom, pakai fungsi isnull().sum(). Dari hasil pengecekan, ternyata kolom CustomerID dan Itemname punya cukup banyak nilai kosong. Karena dua kolom ini penting banget buat analisis, baris yang ada nilai kosong di dua kolom tersebut langsung dibuang pakai fungsi dropna dengan parameter subset=['CustomerID', 'Itemname']. Langkah ini bikin

dataset lebih rapi dan cuma berisi data yang lengkap. Setelah data bersih, dataset ini siap buat dianalisis lebih lanjut, kayak pakai algoritma Apriori buat cari pola pembelian. Proses ini penting biar hasil analisis nantinya akurat dan relevan.

```
[4]: # Hapus baris dengan nilai kosong di CustomerID dan Itemname
data = data.dropna(subset=['CustomerID', 'Itemname'])
```

```
[5]: # 4. Gabungkan Item Berdasarkan BillNo (Transaksi)
data = data.groupby(['BillNo']).agg({
    'Itemname': lambda x: list(x),
    'Quantity': 'sum',
    'Date': 'first',
    'Price': 'sum',
    'CustomerID': 'first',
    'Country': 'first'
}).reset_index()
print("\nData Setelah Penggabungan:")
print(data.head())
```

```
Data Setelah Penggabungan:
   BillNo                                           Itemname  Quantity  \
0  536365  [WHITE HANGING HEART T-LIGHT HOLDER, WHITE MET...        40
1  536366  [HAND WARMER UNION JACK, HAND WARMER RED POLKA...        12
2  536367  [ASSORTED COLOUR BIRD ORNAMENT, POPPY'S PLAYHO...        83
3  536368  [JAM MAKING SET WITH JARS, RED COAT RACK PARIS...        15
4  536369                      [BATH BUILDING BLOCK WORD]         3

              Date  Price  CustomerID         Country
0  01.12.2010 08:26  27.37     17850.0  United Kingdom
1  01.12.2010 08:28   3.70     17850.0  United Kingdom
2  01.12.2010 08:34  58.24     13047.0  United Kingdom
3  01.12.2010 08:34  19.10     13047.0  United Kingdom
4  01.12.2010 08:35   5.95     13047.0  United Kingdom
```

Menghapus Barais dan Nilai Kosong, Menggabungkan Item Berdasarkan Transaksi

Pertama, data dibersihkan dengan menghapus baris yang memiliki nilai kosong di kolom CustomerID dan Itemname, karena data tersebut penting untuk identifikasi pelanggan dan produk yang dibeli. Langkah ini memastikan dataset hanya berisi data yang valid dan lengkap. Selanjutnya, data dikelompokkan berdasarkan BillNo, yaitu ID transaksi, sehingga setiap transaksi direpresentasikan sebagai satu baris data. Dalam proses pengelompokannya, semua produk yang dibeli dalam transaksi tersebut digabung menjadi sebuah daftar pada kolom Itemname. Selain itu, jumlah produk (Quantity) dalam transaksi dijumlahkan, dan informasi lainnya seperti tanggal transaksi (Date), total harga (Price), ID pelanggan (CustomerID), dan negara (Country) diambil berdasarkan nilai pertama dalam grup tersebut.

Setelah data digabung, hasilnya adalah setiap baris menunjukkan satu transaksi lengkap dengan daftar produk yang dibeli dan informasi penting lainnya. Hasil ini membuat dataset lebih mudah dianalisis, terutama untuk algoritma seperti Apriori yang membutuhkan data transaksi dalam format daftar produk.

[6]:
```
# 5. Konversi Kolom Date ke Format Datetime
data['Date'] = pd.to_datetime(data['Date'], format='%d.%m.%Y %H:%M')
data['Year'] = data['Date'].dt.year

# Filter Data Tahun Terbaru
latest_year = data['Year'].max()
data = data[data['Year'] == latest_year]
print(f"\nData Tahun Terbaru ({latest_year}):")
print(data.head())
```

```
Data Tahun Terbaru (2011):
      BillNo                                          Itemname  Quantity  \
1381  539993  [JUMBO BAG PINK POLKADOT, BLUE POLKADOT WRAP, ...       171
1382  540001  [RED HANGING HEART T-LIGHT HOLDER, CERAMIC BOW...       270
1383  540002  [GARDEN METAL SIGN, RED KITCHEN SCALES, VICTOR...       188
1384  540003  [HANGING HEART ZINC T-LIGHT HOLDER, BREAD BIN ...       140
1385  540004                  [ANTIQUE SILVER TEA GLASS ETCHED]        72

                    Date   Price  CustomerID         Country  Year
1381 2011-01-04 10:00:00   38.04     13313.0  United Kingdom  2011
1382 2011-01-04 10:22:00   26.63     18097.0  United Kingdom  2011
1383 2011-01-04 10:23:00   15.82     16656.0  United Kingdom  2011
1384 2011-01-04 10:37:00  131.30     16875.0  United Kingdom  2011
1385 2011-01-04 10:37:00    1.06     13094.0  United Kingdom  2011
```

kolom Date diubah ke format datetime menggunakan fungsi pd.to_datetime. Dengan format ini, data tanggal dapat diolah lebih lanjut, seperti mengekstrak tahun transaksi ke kolom baru Year. Setelah itu, data difilter untuk hanya mencakup transaksi pada tahun terbaru, yaitu 2011, dengan menggunakan nilai maksimum dari kolom Year. Hasil dari langkah ini adalah subset data yang lebih relevan, sehingga analisis pola pembelian yang dilakukan akan fokus pada data terkini.

```
[7]: # 6. One-Hot Encoding untuk Item Transaksi
     te = TransactionEncoder()
     basket_encoded = te.fit(data['Itemname']).transform(data['Itemname'])
     basket_encoded = pd.DataFrame(basket_encoded, columns=te.columns_)
     print("\nData Setelah One-Hot Encoding:")
     print(basket_encoded.head())
```

```
Data Setelah One-Hot Encoding:
   10 COLOUR SPACEBOY PEN  12 COLOURED PARTY BALLOONS  \
0                   False                        False
1                   False                        False
2                   False                        False
3                   False                        False
4                   False                        False

   12 DAISY PEGS IN WOOD BOX  12 EGG HOUSE PAINTED WOOD  \
0                      False                      False
1                      False                      False
2                      False                      False
3                      False                      False
4                      False                      False

   12 HANGING EGGS HAND PAINTED  12 IVORY ROSE PEG PLACE SETTINGS  \
0                         False                             False
```

TransactionEncoder untuk mengubah data daftar item (Itemname) menjadi format One-Hot Encoding. Dalam format ini, setiap item diwakili sebagai kolom dengan nilai True atau False, yang menunjukkan apakah item tersebut ada dalam transaksi tertentu. Transformasi ini penting karena algoritma Apriori hanya dapat bekerja dengan data dalam bentuk matriks biner. Proses ini mempersiapkan data dalam format yang kompatibel untuk analisis pola pembelian.

- Proses feature Dataset

```
[8]: # 7. Menjalankan Algoritma Apriori
     min_support = 0.01  # Set nilai support minimum
     frequent_itemsets = apriori(basket_encoded, min_support=min_support, use_colnames=True)
     print("\nFrequent Itemsets:")
     print(frequent_itemsets)
```

```
Frequent Itemsets:
      support                                        itemsets
0    0.012692                      (10 COLOUR SPACEBOY PEN)
1    0.010070                (12 MESSAGE CARDS WITH ENVELOPES)
2    0.014778                  (12 PENCIL SMALL TUBE WOODLAND)
3    0.016089             (12 PENCILS SMALL TUBE RED RETROSPOT)
4    0.015493                (12 PENCILS SMALL TUBE SKULL)
...       ...                                            ...
1041 0.014480  (ROSES REGENCY TEACUP AND SAUCER, REGENCY CAKE...
1042 0.010070  (WOODEN TREE CHRISTMAS SCANDINAVIAN, WOODEN ST...
1043 0.012990  (ROSES REGENCY TEACUP AND SAUCER, GREEN REGENC...
1044 0.011024  (LUNCH BAG CARS BLUE, LUNCH BAG PINK POLKADOT,...
1045 0.010130  (LUNCH BAG PINK POLKADOT, LUNCH BAG SUKI DESIG...

[1046 rows x 2 columns]
```

algoritma Apriori digunakan untuk menemukan itemset yang sering muncul dalam data transaksi dengan menetapkan nilai minimum support sebesar 0.01 (1%). Data yang telah diubah ke format One-Hot Encoding diproses menggunakan fungsi apriori, menghasilkan tabel dengan kolom support yang menunjukkan frekuensi itemset dalam transaksi dan itemsets yang berisi kombinasi item yang sering muncul. Langkah ini membantu mengidentifikasi pola pembelian yang menjadi dasar untuk analisis lebih lanjut seperti rekomendasi produk atau strategi bundling.

[9]:
```
# 8. Menentukan Aturan Asosiasi
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0, num_itemsets=len(frequent_itemsets))
print("\nAturan Asosiasi:")
print(rules)
```

```
Aturan Asosiasi:
                                          antecedents  \
0                      (60 CAKE CASES DOLLY GIRL DESIGN)
1                      (PACK OF 72 RETROSPOT CAKE CASES)
2                          (60 TEATIME FAIRY CAKE CASES)
3                       (72 SWEETHEART FAIRY CAKE CASES)
4                          (60 TEATIME FAIRY CAKE CASES)
...                                                  ...
1125  (LUNCH BAG  BLACK SKULL., LUNCH BAG RED RETROS...
1126                           (LUNCH BAG PINK POLKADOT)
1127                             (LUNCH BAG SUKI DESIGN)
1128                           (LUNCH BAG  BLACK SKULL.)
1129                           (LUNCH BAG RED RETROSPOT)

                                          consequents  antecedent support  \
0                      (PACK OF 72 RETROSPOT CAKE CASES)            0.019008
1                      (60 CAKE CASES DOLLY GIRL DESIGN)            0.054940
2                       (72 SWEETHEART FAIRY CAKE CASES)            0.034740
3                          (60 TEATIME FAIRY CAKE CASES)            0.026517
4                       (PACK OF 60 DINOSAUR CAKE CASES)            0.034740
...                                                  ...                 ...
1125   (LUNCH BAG SUKI DESIGN, LUNCH BAG PINK POLKADOT)            0.028960
1126  (LUNCH BAG SUKI DESIGN, LUNCH BAG  BLACK SKULL...            0.052854
1127  (LUNCH BAG  BLACK SKULL., LUNCH BAG RED RETROS...            0.053152
1128  (LUNCH BAG SUKI DESIGN, LUNCH BAG RED RETROSPO...            0.059230
1129  (LUNCH BAG SUKI DESIGN, LUNCH BAG  BLACK SKULL...            0.072280

      consequent support   support  confidence       lift  representativity  \
0                0.054940  0.010487    0.551724  10.042337               1.0
1                0.019008  0.010487    0.190889  10.042337               1.0
2                0.026517  0.011620    0.334477  12.613911               1.0
3                0.034740  0.011620    0.438202  12.613911               1.0
4                0.029436  0.012215    0.351630  11.945438               1.0
...                   ...       ...         ...        ...               ...
1125             0.021213  0.010130    0.349794  16.489458               1.0
1126             0.014897  0.010130    0.191657  12.865569               1.0
1127             0.017161  0.010130    0.190583  11.105428               1.0
1128             0.014897  0.010130    0.171026  11.480644               1.0
1129             0.013109  0.010130    0.140148  10.690774               1.0
```

| | leverage | conviction | zhangs_metric | jaccard | certainty | kulczynski |
|---|---|---|---|---|---|---|
| 0 | 0.009443 | 2.108211 | 0.917869 | 0.165258 | 0.525664 | 0.371307 |
| 1 | 0.009443 | 1.212432 | 0.952766 | 0.165258 | 0.175211 | 0.371307 |
| 2 | 0.010698 | 1.462734 | 0.953859 | 0.234094 | 0.316349 | 0.386340 |
| 3 | 0.010698 | 1.718164 | 0.945802 | 0.234094 | 0.417983 | 0.386340 |
| 4 | 0.011193 | 1.496928 | 0.949263 | 0.235092 | 0.331965 | 0.383305 |
| ... | ... | ... | ... | ... | ... | ... |
| 1125 | 0.009516 | 1.505349 | 0.967370 | 0.252976 | 0.335702 | 0.413661 |
| 1126 | 0.009343 | 1.218670 | 0.973739 | 0.175801 | 0.179433 | 0.435829 |
| 1127 | 0.009218 | 1.214255 | 0.961035 | 0.168317 | 0.176450 | 0.390430 |
| 1128 | 0.009248 | 1.188340 | 0.970372 | 0.158287 | 0.158490 | 0.425513 |
| 1129 | 0.009182 | 1.147745 | 0.977085 | 0.134600 | 0.128727 | 0.456438 |

[1130 rows x 14 columns]

membuat aturan asosiasi menggunakan metrik lift dengan nilai minimum 1.0 untuk memastikan hubungan antar item kuat. Hasilnya adalah tabel aturan yang menunjukkan antecedents (item awal), consequents (item terkait), dan metrik seperti support, confidence, serta lift. Contohnya, item "60 CAKE CASES DOLLY GIRL DESIGN" memiliki hubungan kuat dengan "PACK OF 72 RETROSPOT CAKE CASES". Informasi ini berguna untuk strategi bundling produk atau rekomendasi belanja.

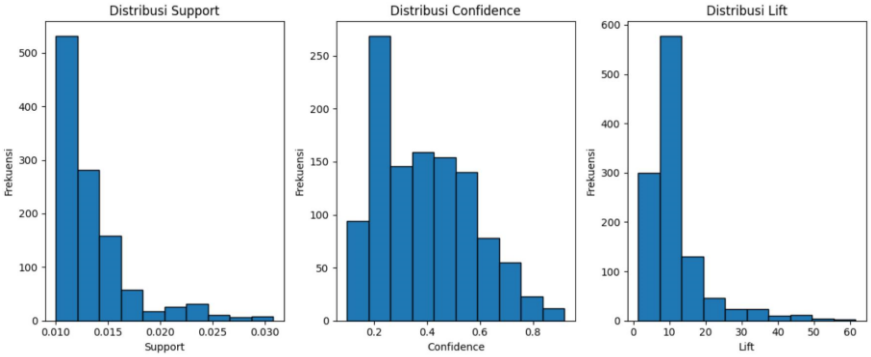
```
# 10. Visualisasi Distribusi Support, Confidence, dan Lift
plt.figure(figsize=(12, 5))

# Support
plt.subplot(1, 3, 1)
plt.hist(rules['support'], bins=10, edgecolor='black')
plt.title('Distribusi Support')
plt.xlabel('Support')
plt.ylabel('Frekuensi')

# Confidence
plt.subplot(1, 3, 2)
plt.hist(rules['confidence'], bins=10, edgecolor='black')
plt.title('Distribusi Confidence')
plt.xlabel('Confidence')
plt.ylabel('Frekuensi')

# Lift
plt.subplot(1, 3, 3)
plt.hist(rules['lift'], bins=10, edgecolor='black')
plt.title('Distribusi Lift')
plt.xlabel('Lift')
plt.ylabel('Frekuensi')

plt.tight_layout()
plt.show()
```



membuat histogram untuk visualisasi distribusi metrik support, confidence, dan lift dari aturan asosiasi. Grafik pertama menunjukkan bahwa sebagian besar itemset memiliki nilai support rendah, artinya jarang muncul. Grafik kedua memperlihatkan distribusi confidence, di mana sebagian besar aturan memiliki confidence antara 0.2 hingga 0.6. Grafik terakhir menunjukkan lift, dengan banyak aturan memiliki lift yang tinggi, menandakan hubungan kuat antar item.

- **Performa Model**

  Model aturan asosiasi menunjukkan performa yang baik dengan banyak aturan memiliki lift di atas 1, menandakan hubungan antar produk yang signifikan. Sebagian besar aturan memiliki support rendah, namun pola yang ditemukan tetap relevan untuk analisis, terutama untuk produk dengan hubungan kuat.

- **Diskusi Hasil dan Kesimpulan**

  Hasil analisis menunjukkan pola pembelian yang signifikan di antara produk tertentu.
  Misalnya:
  - Contoh 1:
    - Antecedents: 60 CAKE CASES DOLLY GIRL DESIGN
    - Consequents: PACK OF 72 RETROSPOT CAKE CASES
    - Support: 1.05% dari total transaksi
    - Lift: 10.04 (hubungan 10 kali lebih kuat dari ekspektasi independen)
  - Contoh 2:
    - Antecedents: 72 SWEETHEART FAIRY CAKE CASES
    - Consequents: 60 TEATIME FAIRY CAKE CASES
    - Support: 1.16%
    - Lift: 12.61 (hubungan signifikan di antara produk ini)

  Aturan-aturan ini menunjukkan bahwa produk-produk tertentu sering dibeli bersama, meskipun frekuensinya tidak terlalu tinggi. Lift yang signifikan memperkuat hubungan antar produk dalam aturan.

- **Implementasi**
  - Rekomendasi Produk:
    Jika pelanggan membeli 60 CAKE CASES DOLLY GIRL DESIGN, kasir atau staff dapat merekomendasikan PACK OF 72 RETROSPOT CAKE CASES untuk meningkatkan peluang pembelian tambahan. Dalam platform e-commerce, aturan ini dapat diterapkan melalui fitur "produk yang sering dibeli bersama".
  - Strategi Bundling :
    Produk seperti 72 SWEETHEART FAIRY CAKE CASES dan 60 TEATIME FAIRY CAKE CASES dapat dijual sebagai satu paket dengan diskon khusus untuk mendorong penjualan.
  - Promosi yang Ditargetkan:
    Menggunakan aturan asosiasi untuk menentukan pasangan produk yang sering dibeli, sehingga promosi dapat difokuskan pada produk-produk yang memiliki hubungan kuat.
  - Manajemen Stok:
    Aturan asosiasi membantu memprediksi kebutuhan stok produk yang sering dibeli bersama, mengurangi risiko kekurangan stok produk yang terkait.

- **Kesimpulan**

Model berhasil mengidentifikasi hubungan kuat antar produk, seperti kombinasi item dengan lift signifikan. Implementasi hasil ini, seperti strategi rekomendasi atau bundling produk, dapat meningkatkan penjualan dan kepuasan pelanggan. Langkah berikutnya adalah memvalidasi aturan pada data transaksi baru untuk memastikan relevansi pola pembelian dalam skenario nyata.

Link Github : https://github.com/Tataaa2/DataMining/tree/main/DMdeploy

Link Deploy : https://datamining-zeoowfklfyfjdz94k283ec.streamlit.app/