

Penerapan Algoritma Apriori dalam Menyusun Rekomendasi Paket Menu Restoran Berdasarkan Pola Transaksi Pelanggan

Dosen pembimbing :

Dewa Putu Wiadnyana Putra, S.Pd., M.Si



Oleh : Windi Pangesti



Latar belakang

Dalam perkembangan industri kuliner yang pesat, para pemilik restoran perlu menghadapi persaingan ketat dengan menyusun strategi yang tepat. Kenaikan aktivitas penjualan meningkatkan volume data transaksi, dan dengan menerapkan algoritma Apriori, restoran dapat mengidentifikasi pola kombinasi produk yang sering dibeli bersamaan oleh pelanggan. Informasi ini dapat digunakan untuk menyusun rekomendasi paket menu yang membantu pelanggan dalam pengambilan keputusan saat memesan makanan, dengan harapan meningkatkan omset penjualan melalui penawaran paket yang menarik.



Landasan Teori



Aturan asosiasi adalah ekspresi yang berbentuk implikasi $X \Rightarrow Y$ dengan X dan Y merupakan himpunan item dan $X \cap Y = \{ \}$. Ekspresi tersebut mempunyai makna sebab akibat dalam transaksi, yaitu jika transaksi memuat semua item dalam X maka transaksi memuat semua item dalam Y .

Algoritma Apriori merupakan algoritma yang sangat terkenal untuk menemukan pola frekuensi tinggi, Algoritma Apriori ini termasuk jenis aturan asosiasi pada data mining. Pola frekuensi tinggi adalah pola-pola item di dalam suatu basis data yang memiliki frekuensi atau support.

Support merupakan suatu ukuran yang menunjukkan seberapa besar tingkat dominasi dari suatu barang atau itemset dari keseluruhan transaksi yang ada.

Confidence adalah suatu ukuran yang menunjukkan hubungan kondisional antar dua barang, misalnya seberapa sering produk D dibeli jika orang membeli produk C

Contoh :

$$X \Rightarrow Y [\text{support} = 1\% , \text{confidence} = 50\%]$$

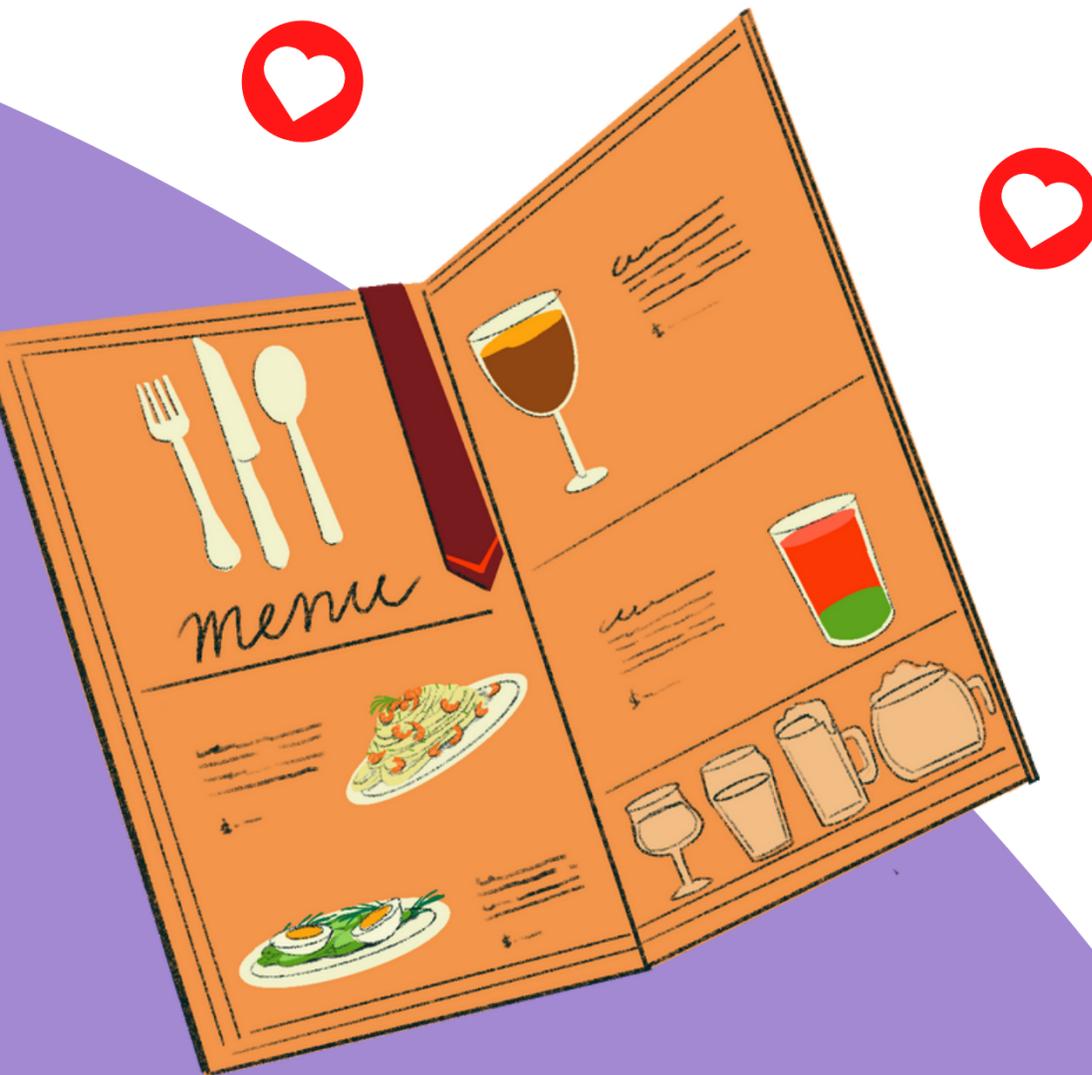
nilai confidence 50% artinya jika seorang pelanggan membeli x maka kemungkinan 50% pelanggan tersebut juga akan membeli Y. sedangkan support 1% artinya 1% dari seluruh transaksi yang diperiksa menunjukkan bahwa X dan Y dibeli secara bersamaan)

Tujuan Analisis :

- Memberikan informasi mengenai produk yang sering dibeli bersamaan oleh pelanggan yang dapat dimanfaatkan untuk **menyusun rekomendasi paket menu makanan**.
- Menerapkan **data mining association rule dengan algoritma apriori** untuk mendapatkan rekomendasi pasangan produk.

Metode Analisis:

Proses **data mining menggunakan metode Association Rule dengan algoritma Apriori**. Algoritma ini cocok untuk analisis transaksi pembelian, **menemukan aturan asosiasi barang yang sering dibeli bersama**. Hasilnya dapat digunakan sebagai sumber informasi untuk pengembangan promosi, pemahaman pola pembelian konsumen, dan pembuatan aturan asosiasi menu untuk pengambilan keputusan.




```
0d len(df)
11000
```

Jumlah baris pada tabel transaksi
11.000

```
0d [29] df['Transaction_ID'].nunique()
11000
```

Jumlah ID unique Transaksi
11.000

```
0d [30] df['Customer_ID'].nunique()
1081
```

Jumlah unique ID Customer
1081

```
[ ] # Checking How many unique menu in the restaurant
pd.concat([df['M1'],df['M2'],df['M3'],df['M4'],df['M5'],df['M6'],df['M7'],df['M8'],df['M9']]).unique()

array(['Bhuna - Chicken', 'Plain Rice', nan, 'House white wine 75cl',
      'Chicken Tikka Balti', 'Special Fried Rice', 'Curry',
      'Lamb Biryani', 'Vindaloo - Prawn', 'Korma - Lamb'], dtype=object)
```

Ada 9 menu yang disediakan restoran, yaitu

- 1. Bhuna – Chicken
- 2. Plain Rice
- 3. House white wine 75cl
- 4. Chicken Tikka Balti
- 5. Special Fried Rice
- 6. Curry
- 7. Lamb Biryani
- 8. Vindaloo – Prawn
- 9. Korma – Lamb

```
[31] # Checking null value in dataframe
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11000 entries, 0 to 10999
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Transaction_ID   11000 non-null  object
1   Customer_ID     11000 non-null  object
2   M1              11000 non-null  object
3   M2              11000 non-null  object
4   M3              9447 non-null   object
5   M4              7699 non-null   object
6   M5              5721 non-null   object
7   M6              3429 non-null   object
8   M7              1399 non-null   object
9   M8              304 non-null    object
10  M9              34 non-null     object
dtypes: object(11)
memory usage: 945.4+ KB
```

Tidak ada data null pada masing masing kolom

Membuat Fungsi untuk mengelompokkan menu pada masing-masing kolom

```
[35] # Make function to categorize menu
# Bhuna - Chicken
def conditions1(df):
    if (df["M1"] == "Bhuna - Chicken") or (df["M2"] == "Bhuna - Chicken") or (df["M3"] == "Bhuna - Chicken") or (df["M4"] == "Bhuna - Chicken") or (df["M5"] == '
        return "Bhuna - Chicken"
    else:
        return np.NaN

# Chicken Tikka Balti
def conditions2(df):
    if (df["M1"] == "Chicken Tikka Balti") or (df["M2"] == "Chicken Tikka Balti") or (df["M3"] == "Chicken Tikka Balti") or (df["M4"] == "Chicken Tikka Balti")
        return "Chicken Tikka Balti"
    else:
        return np.NaN

# Curry
def conditions3(df):
    if (df["M1"] == "Curry") or (df["M2"] == "Curry") or (df["M3"] == "Curry") or (df["M4"] == "Curry") or (df["M5"] == "Curry") or (df["M6"] == "Curry") or (df
        return "Curry"
    else:
        return np.NaN

# House white wine 75cl
def conditions4(df):
    if (df["M1"] == "House white wine 75cl") or (df["M2"] == "House white wine 75cl") or (df["M3"] == "House white wine 75cl") or (df["M4"] == "House white wine
        return "House white wine 75cl"
```

Mengaplikasikan fungsi pada masing-masing menu

```
[36] # Add new column for 9 menus with apply the functions
df['Bhuna - Chicken'] = df.apply(conditions1, axis=1)
df['Chicken Tikka Balti'] = df.apply(conditions2, axis=1)
df['Curry'] = df.apply(conditions3, axis=1)
df['House white wine 75cl'] = df.apply(conditions4, axis=1)
df['Korma - Lamb'] = df.apply(conditions5, axis=1)
df['Lamb Biryani'] = df.apply(conditions6, axis=1)
df['Plain Rice'] = df.apply(conditions7, axis=1)
df['Special Fried Rice'] = df.apply(conditions8, axis=1)
df['Vindaloo - Prawn'] = df.apply(conditions9, axis=1)
```

Tampilan tabel setelah dikelompokkan sesuai menu

```
[37] # Make new dataframe that contains the menu colomn
data = df[['Bhuna - Chicken', 'Chicken Tikka Balti', 'Curry', 'House white wine 75cl', 'Korma - Lamb', 'Lamb Biryani', 'Plain Rice', 'Special Fried Rice', 'Vindaloo - I
data
```

	Bhuna - Chicken	Chicken Tikka Balti	Curry	House white wine 75cl	Korma - Lamb	Lamb Biryani	Plain Rice	Special Fried Rice	Vindaloo - Prawn
0	Bhuna - Chicken	NaN	NaN	NaN	Korma - Lamb	NaN	NaN	NaN	NaN
1	Bhuna - Chicken	NaN	Curry	NaN	Korma - Lamb	NaN	Plain Rice	Special Fried Rice	NaN
2	NaN	Chicken Tikka Balti	NaN	NaN	NaN	NaN	Plain Rice	NaN	NaN
3	Bhuna - Chicken	NaN	Curry	House white wine 75cl	NaN	NaN	NaN	Special Fried Rice	Vindaloo - Prawn
4	NaN	NaN	NaN	NaN	Korma - Lamb	NaN	Plain Rice	NaN	Vindaloo - Prawn
...
10995	NaN	NaN	NaN	House white wine 75cl	NaN	NaN	Plain Rice	Special Fried Rice	Vindaloo - Prawn
10996	Bhuna - Chicken	Chicken Tikka Balti	Curry	NaN	NaN	Lamb Biryani	NaN	Special Fried Rice	NaN
10997	NaN	NaN	NaN	House white wine 75cl	Korma - Lamb	NaN	Plain Rice	Special Fried Rice	NaN
10998	Bhuna - Chicken	Chicken Tikka Balti	Curry	NaN	Korma - Lamb	NaN	Plain Rice	Special Fried Rice	Vindaloo - Prawn
10999	NaN	NaN	Curry	NaN	Korma - Lamb	NaN	NaN	Special Fried Rice	Vindaloo - Prawn

11000 rows × 9 columns

Memberikan variabel dummy untuk melakukan association rules

```
[39] # tidy data for association rules
data_dummy = pd.get_dummies(data)
data_dummy.columns = ['Bhuna - Chicken', 'Chicken Tikka Balti', 'Curry', 'House white wine 75cl', 'Korma - Lamb', 'Lamb Biryani', 'Plain Rice', 'Special Fried Rice',
data_dummy
```

	Bhuna - Chicken	Chicken Tikka Balti	Curry	House white wine 75cl	Korma - Lamb	Lamb Biryani	Plain Rice	Special Fried Rice	Vindaloo - Prawn
0	1	0	0	0	1	0	0	0	0
1	1	0	1	0	1	0	1	1	0
2	0	1	0	0	0	0	1	0	0
3	1	0	1	1	0	0	0	1	1
4	0	0	0	0	1	0	1	0	1
...
10995	0	0	0	1	0	0	1	1	1
10996	1	1	1	0	0	1	0	1	0
10997	0	0	0	1	1	0	1	1	0
10998	1	1	1	0	1	0	1	1	1
10999	0	0	1	0	1	0	0	1	1

11000 rows x 9 columns

Keterangan :

1 - item dibeli oleh pelanggan

0 - item tidak dibeli oleh pelanggan

membentuk itemset dengan menggunakan algoritma apriori dengan menetapkan minimum supportnya = 0.05. diperoleh itemset sebagai berikut :

```
[40] # membangun model
frq_items = apriori(data_dummy, min_support=0.05, use_colnames=True)
frq_items['length'] = frq_items['itemsets'].apply(lambda x: len(x))
frq_items.sort_values(by="support", ascending=False)
```

	support	itemsets	length
6	0.463091	(Plain Rice)	1
2	0.462909	(Curry)	1
7	0.462000	(Special Fried Rice)	1
5	0.457727	(Lamb Biryani)	1
3	0.456455	(House white wine 75cl)	1
...
112	0.094636	(Korma - Lamb, Vindaloo - Prawn, House white w...	3
79	0.094455	(Korma - Lamb, House white wine 75cl, Chicken ...	3
94	0.094455	(Korma - Lamb, House white wine 75cl, Curry)	3
62	0.094091	(Vindaloo - Prawn, House white wine 75cl, Bhun...	3
53	0.092455	(Korma - Lamb, Curry, Bhuna - Chicken)	3
...



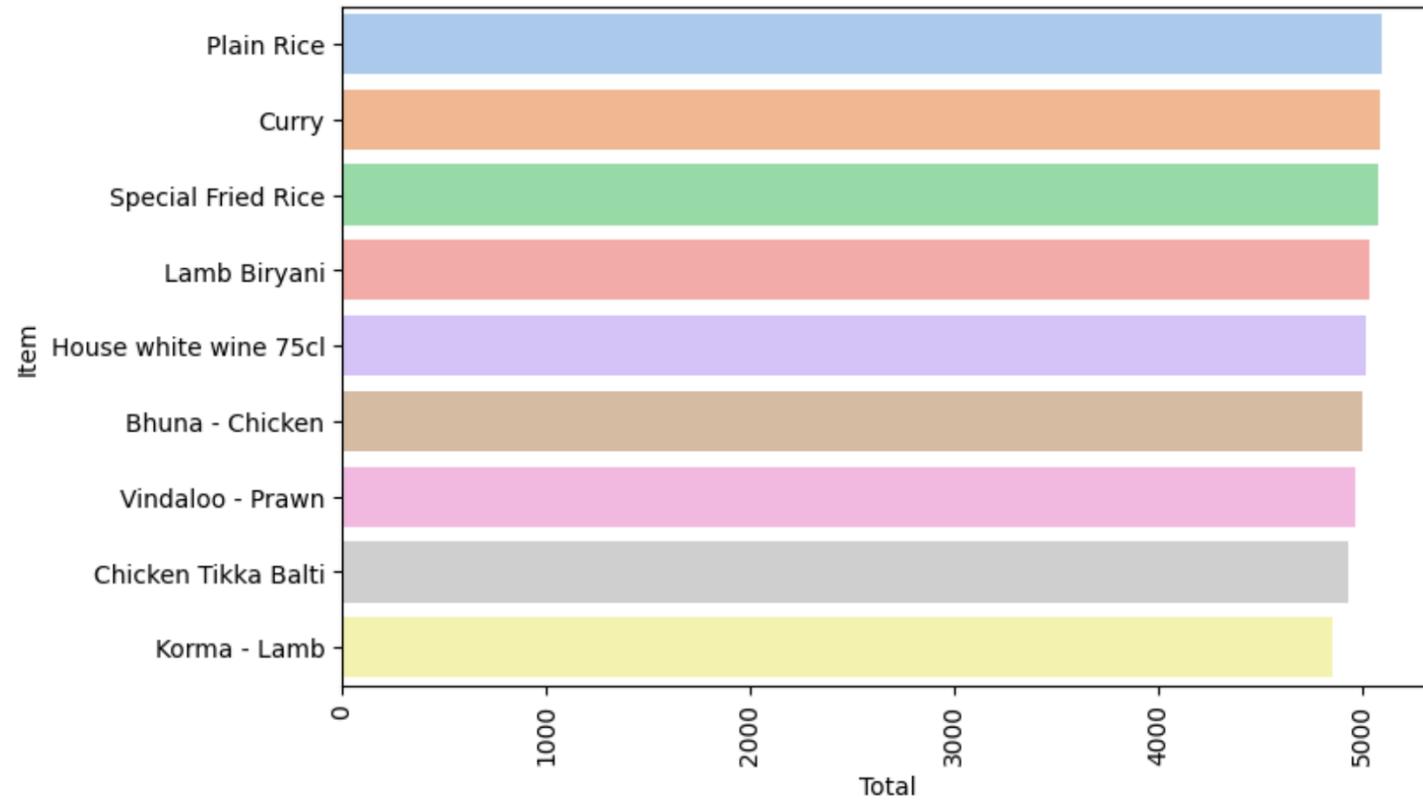
melakukan aturan asosiasi untuk menampilkan 5 pasangan item tertinggi yang memenuhi aturan yang diurutkan berdasarkan nilai confidence dan lift :

```
[41] # mengumpulkan aturan yang memenuhi dataframe
rules = association_rules(freq_items, metric="lift", min_threshold=1)
rules = rules.sort_values(['confidence', 'lift'], ascending=[False, False])
rules.head(5)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
156	(Korma - Lamb, Bhuna - Chicken)	(Special Fried Rice)	0.205091	0.462000	0.102091	0.497784	1.077454	0.007339	1.071252	0.090433
410	(Korma - Lamb, House white wine 75cl)	(Special Fried Rice)	0.205909	0.462000	0.101727	0.494040	1.069350	0.006597	1.063325	0.081669
288	(Lamb Biryani, Chicken Tikka Balti)	(Plain Rice)	0.206909	0.463091	0.101636	0.491213	1.060726	0.005819	1.055272	0.072185
352	(Korma - Lamb, Curry)	(Special Fried Rice)	0.202818	0.462000	0.099545	0.490811	1.062362	0.005843	1.056583	0.073636
136	(House white wine 75cl, Bhuna - Chicken)	(Special Fried Rice)	0.206273	0.462000	0.100909	0.489202	1.058879	0.005611	1.053255	0.070056

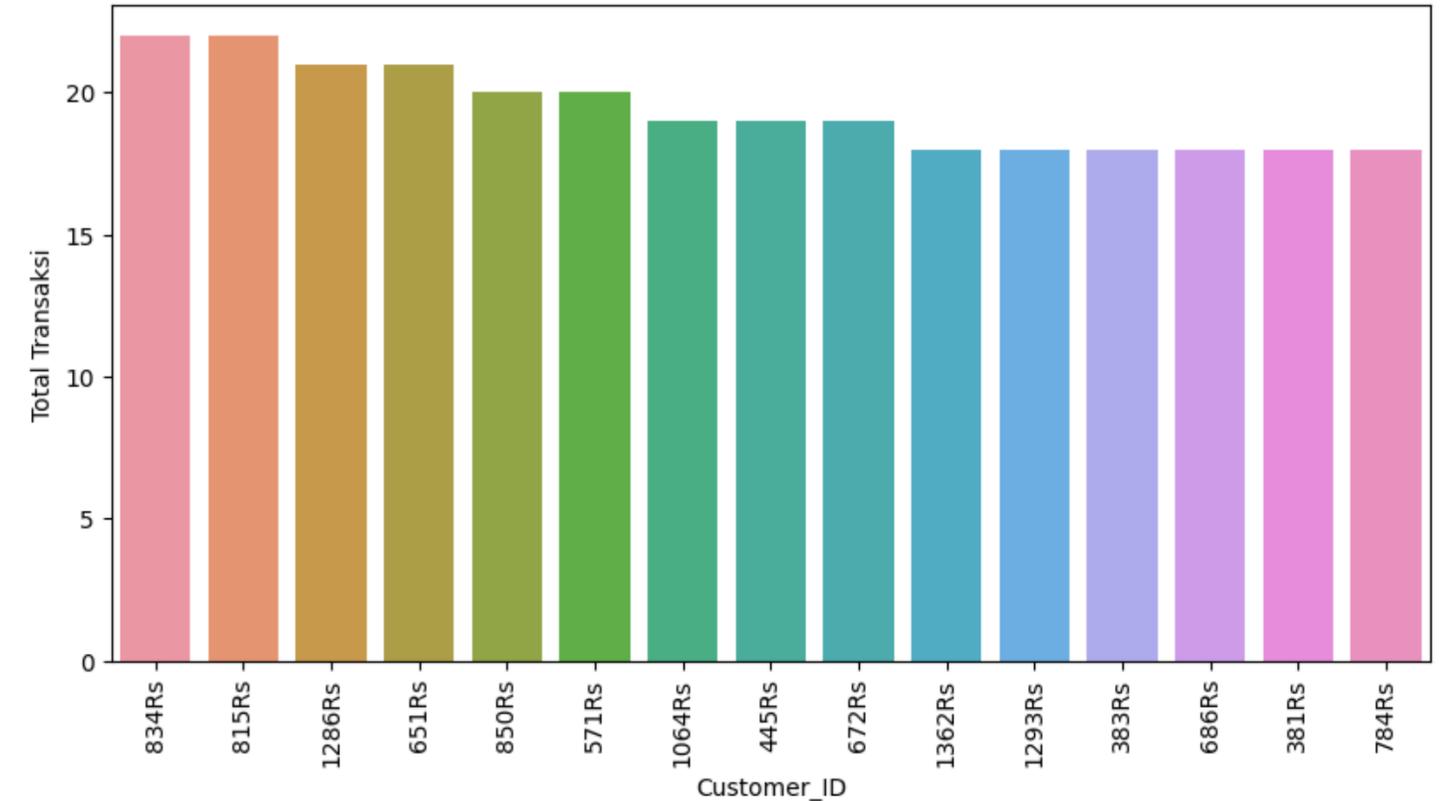
Hasil di atas menampilkan 5 paket menu teratas yang direkomendasikan

Total sales of different Item



Visualisasi di atas menampilkan bahwa menu plain rice memiliki total penjualan tertinggi. Selain itu, dapat dilihat bahwa total penjualan masing-masing menu tidak memiliki perbedaan yang signifikan.

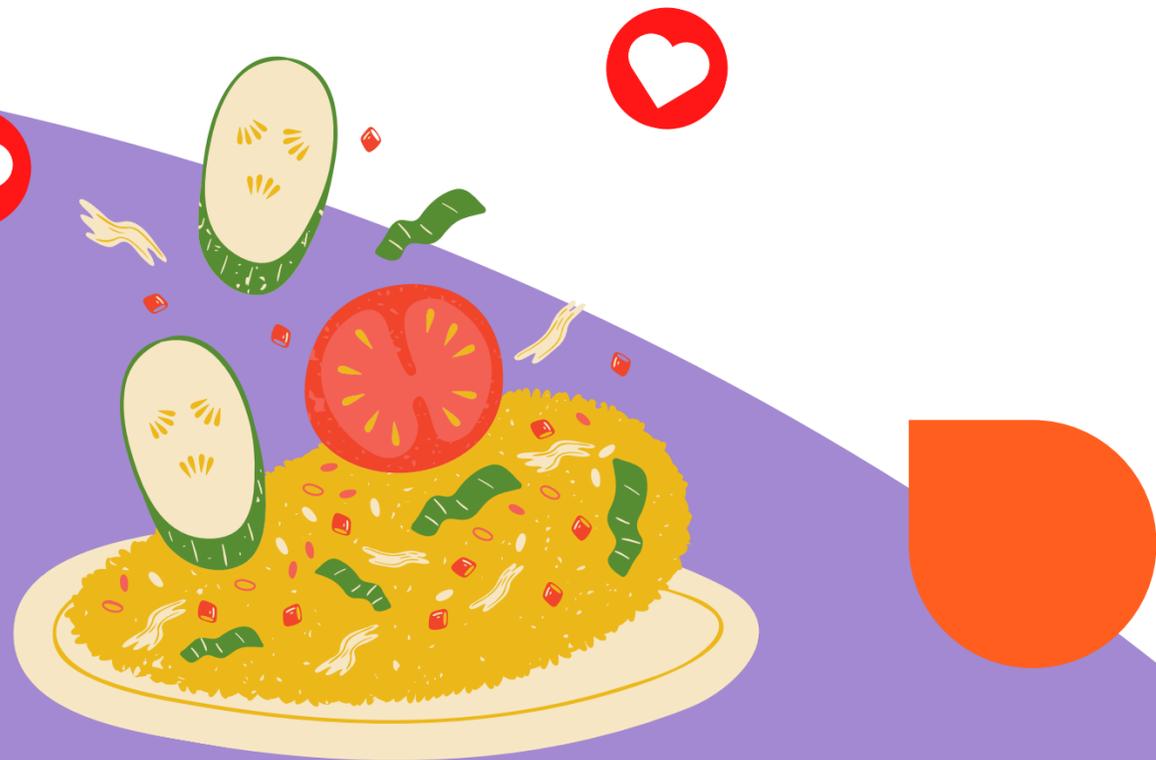
Total Transaksi per customer



Dapat dilihat dari bar chart di atas bahwa pelanggan yang memiliki total transaksi tertinggi adalah pelanggan dengan ID 834Rs.

Kesimpulan

Berdasarkan hasil Analisis, Algoritma apriori berhasil digunakan untuk menghitung dan menemukan aturan asosiasi dengan menggunakan nilai minimum support = 0.05 dan minimum threshold = 1. Dan menghasilkan hasil rekomendasi paket menu makanan sebagai berikut



No	Rekomendasi Menu
1	Bhuna - Chicken, Korma - Lamb, Special fried rice
2	House white wine 75cl, Korma - Lamb, Special fried rice
3	Lamb biryani, Chicken tikka balti, Plain rice
4	Curry, Korma - Lamb, Special fried rice
5	House white wine 75cl, Bhuna chicken, Special fried rice

5 pasangan menu diatas adalah hasil rekomendasi terbaik berdasarkan nilai confidence dan lift nya.