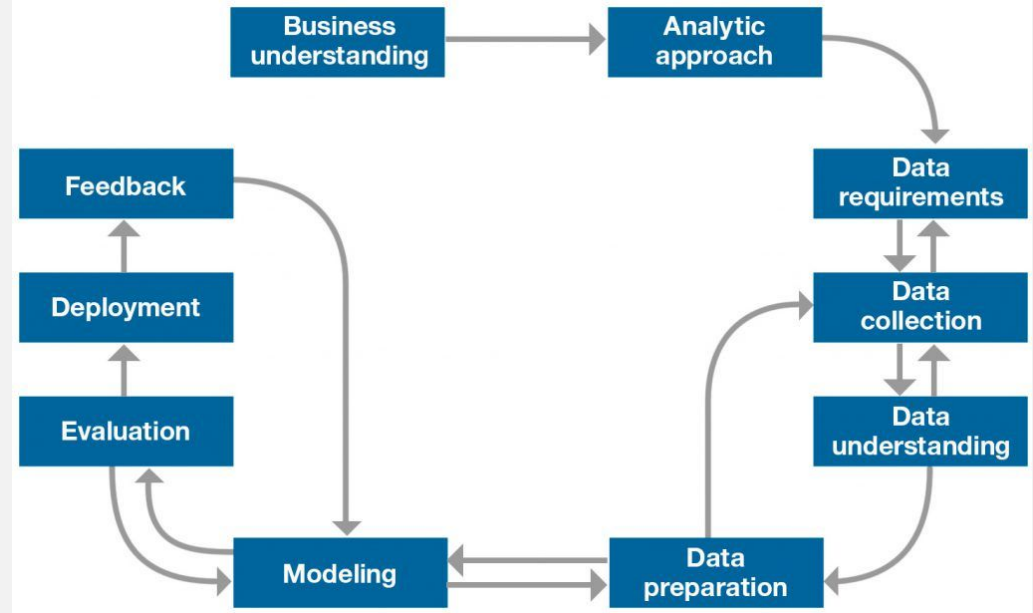


# **Sistem Rekomendasi Menggunakan Model Association Rule dengan Algoritma Apriori**

**Project Akhir PSD**



# Metodologi Sains Data



01



# Business Understanding

Melakukan Klarifikasi terhadap masalah

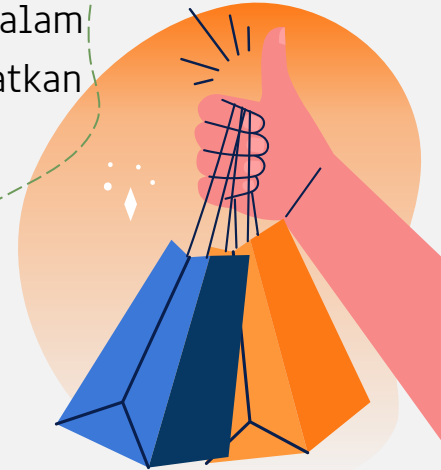
# Business Understanding

## Goals :

memberikan rekomendasi dengan harapan membantu konsumen mengerucutkan pilihannya

## Objectives:

Mengembangkan model yang dapat digunakan dalam implementasi sistem rekomendasi agar pengguna mendapatkan rekomendasi yang sesuai.



02



# Analytic Approach

Menentukan metode analitik

# Analytic Approach



**Predictive  
Analytic**



**Prescriptive  
Analytic**



03



# Data Requirements

Menentukan data

# Data Requirements

Diperlukan data itemset transaksi-transaksi yang dilakukan oleh setiap pelanggan.





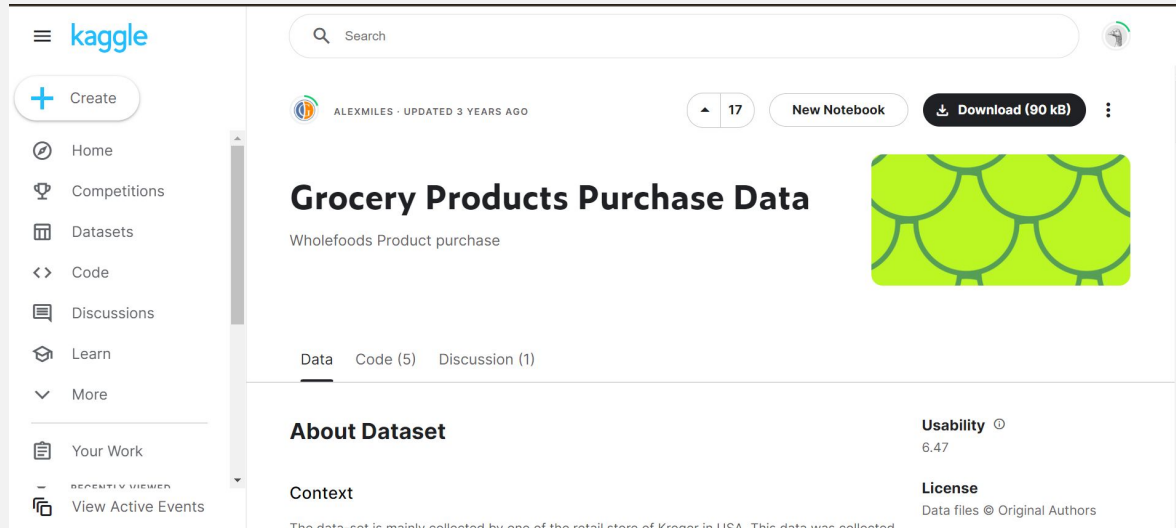
04

# Data Collection

Pengumpulan Data



# Data Collections



The screenshot displays the Kaggle website interface. On the left is a navigation sidebar with the Kaggle logo, a 'Create' button, and links to Home, Competitions, Datasets, Code, Discussions, Learn, More, Your Work, and View Active Events. The main content area features a search bar, a user profile for 'ALEXSMILES' (updated 3 years ago), and a '17' badge. Below this is the dataset title 'Grocery Products Purchase Data' with a subtitle 'Wholefoods Product purchase' and a green patterned image. A tab bar shows 'Data', 'Code (5)', and 'Discussion (1)'. The 'About Dataset' section includes a 'Context' heading and a description: 'The dataset is mainly collected by one of the retail store of Kroger in USA. This data was collected...'. On the right, the 'Usability' score is 6.47 and the 'License' is 'Data files © Original Authors'. A 'Download (90 kB)' button is also visible.

**Kaggle**

Search

ALEXSMILES · UPDATED 3 YEARS AGO

17

New Notebook

Download (90 kB)

## Grocery Products Purchase Data

Wholefoods Product purchase

Data Code (5) Discussion (1)

### About Dataset

#### Context

The dataset is mainly collected by one of the retail store of Kroger in USA. This data was collected...

**Usability** 6.47

**License**  
Data files © Original Authors

# Data Collections

## ▼ Import dataset dengan kaggle API



```
from google.colab import files  
files.upload()
```



Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving kaggle (1).json to kaggle (1) (2).json

```
{'kaggle (1).json': b'{"username": "alyafitrinurhaliza", "key": "29085255418d483563af9a6546df0a4a"}'}
```

```
[ ] ! mkdir ~/.kaggle
```

mkdir: cannot create directory '/root/.kaggle': File exists

05



# Data Understanding

Memahami dan Menganalisis Data

# Data Understanding

## 1. Ukuran Dataset

Memiliki 9835 baris dan 32 kolom

## 2. Atribut Data

Nama Daftar kolom yang berada dalam dataset

## 3. Daftar Tipe data

Semua Kolom memiliki tipe data object

### Informasi Ukuran Dataset

```
[14] print('Ukuran data:', df.shape)
```

```
Ukuran data: (9835, 32)
```

### Informasi Atribut Data

```
print('Daftar kolom:', df.columns)
```

```
Daftar kolom: Index(['Product 1', 'Product 2', 'Product 3', 'Product 4', 'Product 5',  
                    'Product 6', 'Product 7', 'Product 8', 'Product 9', 'Product 10',  
                    'Product 11', 'Product 12', 'Product 13', 'Product 14', 'Product 15',  
                    'Product 16', 'Product 17', 'Product 18', 'Product 19', 'Product 20',  
                    'Product 21', 'Product 22', 'Product 23', 'Product 24', 'Product 25',  
                    'Product 26', 'Product 27', 'Product 28', 'Product 29', 'Product 30',  
                    'Product 31', 'Product 32'],  
                    dtype='object')
```

```
[16] print('Daftar tipe data:\n' + str(df.dtypes))
```

```
Daftar tipe data:  
Product 1    object  
Product 2    object  
Product 3    object  
Product 4    object  
Product 5    object  
Product 6    object  
Product 7    object  
Product 8    object  
Product 9    object  
Product 10   object  
Product 11   object  
Product 12   object  
Product 13   object  
Product 14   object  
Product 15   object  
Product 16   object  
Product 17   object  
Product 18   object  
Product 19   object  
Product 20   object  
Product 21   object  
Product 22   object  
Product 23   object  
Product 24   object  
Product 25   object  
Product 26   object  
Product 27   object  
Product 28   object  
Product 29   object  
Product 30   object  
Product 31   object  
Product 32   object
```

# Data Understanding

```
[ ] 1 df.head(5)
```

	Product 1	Product 2	Product 3	Product 4	Product 5	Product 6	Product 7	Product 8	Product 9	Product 10	...	Product 23	Product 24	Product 25	Product 26	Product 27	Product 28	Product 29	Product 30	Product 31	Product 32
0	citrus fruit	semi-finished bread	margarine	ready soups	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	tropical fruit	yogurt	coffee	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	whole milk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	pip fruit	yogurt	cream cheese	meat spreads	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	other vegetables	whole milk	condensed milk	long life bakery product	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows x 32 columns

# Data Understanding

## 4. Exploratory Data Analysis (EDA)

- Analisis dilakukan hanya dengan melihat distribusi data pada jumlah produk yang dibeli pada setiap transaksi dan disimpan dalam kolom baru dengan nama *count*.

```
✓ [99] 1 df_count['Count'].describe()  
0s  
  
count    9835.000000  
mean      4.409456  
std       3.589385  
min       1.000000  
25%      2.000000  
50%      3.000000  
75%      6.000000  
max      32.000000  
Name: Count, dtype: float64
```

# Data Understanding

## 4. Exploratory Data Analysis (EDA)

- Analisis terhadap jumlah pembelian yang telah dilakukan pada setiap item pada data transaksi

```
[90] 1 occurrences = occurrences.sort_values(by='count', ascending=False)
      2 occurrences
```

	item	count
7	whole milk	2513
11	other vegetables	1903
17	rolls/buns	1809
31	soda	1715
5	yogurt	1372
...	...	...
167	kitchen utensil	4
164	bags	4
168	preservation products	2
166	sound storage medium	1
159	baby food	1

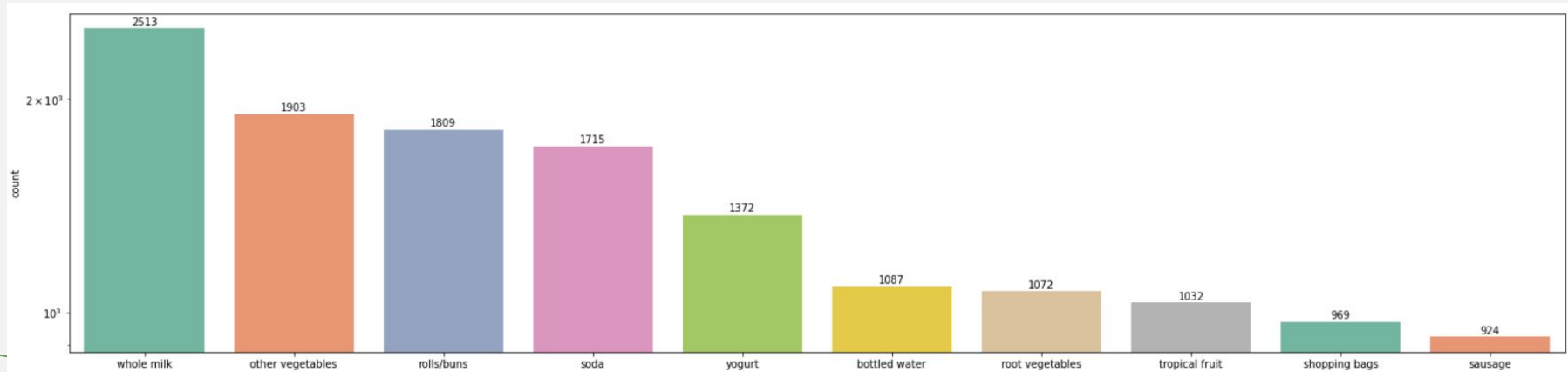
169 rows x 2 columns



# Data Understanding

## 4. Exploratory Data Analysis (EDA)

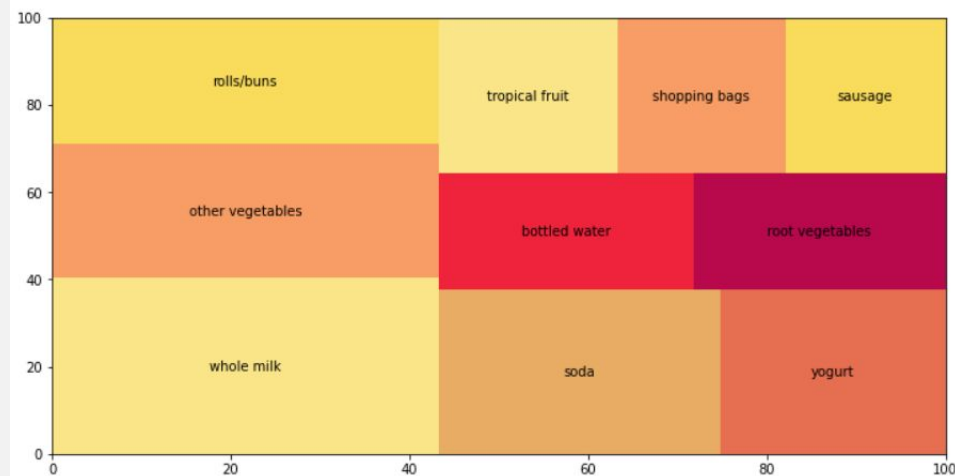
- Visualisasi data dengan bar chart atau bar plot 10 item dengan jumlah pembelian tertinggi dilakukan menggunakan library matplotlib dan seaborn.



# Data Understanding

## 4. Exploratory Data Analysis (EDA)

- visualisasi dengan treemap 10 item dengan jumlah pembelian tertinggi, kami menggunakan library squarify.



06



# Data Preparations

Melakukan Pra-Preproses Data

# Data Preparation

1. Terdapat 271353 sel data yang bernilai None tetapi tidak perlu dilakukan pre proses handling missing value.
2. Terdapat 2824 data yang bernilai sama atau terdapat duplikat data.
3. Melakukan preprocessing text yaitu case folding

```
for i in df:  
    df[i] = df[i].str.lower()
```

4. Melakukan encode pada data transaksi ke dari data frame ke list of lists

```
[ ] 1 from collections import defaultdict  
2 item_counts = defaultdict(int) # per item occurrences  
3 list_cart = []  
4  
5 for x in range(len(df)):  
6     cart = df.iloc[x].dropna().to_list() # makes item lists  
7     for items in cart:  
8         for item in items.split(", "):  
9             item_counts[item] += 1  
10    list_cart.append(cart)
```

# Data Preparation

5. Melakukan grouping berdasarkan nama item yang sama dan melakukan checking pada masing-masing item

```
[ ] 1 occurences = pd.DataFrame.from_dict(item_counts, orient='index', columns=['count']).reset_index()
    2 occurences = occurences.rename(columns={'index':'item'})
    3 occurences = occurences.sort_values(by='item')
    4 occurences
```

	item	count
16	abrasive cleaner	35
61	artif. sweetener	32
144	baby cosmetics	6
159	baby food	1
164	bags	4
...	...	...
23	white bread	414
94	white wine	187
7	whole milk	2513
5	yogurt	1372
64	zwieback	68

169 rows × 2 columns

# Data Preparation

6. Setelah menjadi list of lists, data di-encode dengan kode berikut:

```
te = TransactionEncoder()  
te_ary = te.fit(list_cart).transform(list_cart)  
te_ary
```

7. Hasil dari data yang telah di encode adalah NumPy array sebagai berikut:

```
array([[False, False, False, ..., False, False, False],  
       [False, False, False, ..., False,  True, False],  
       [False, False, False, ...,  True, False, False],  
       ...,  
       [False, False, False, ..., False,  True, False],  
       [False, False, False, ..., False, False, False],  
       [False, False, False, ..., False, False, False]])
```

# Data Preparation

8. Format diubah kembali menjadi dataframe dengan kode sebagai berikut:

```
te_df = pd.DataFrame(te_ary, columns=te.columns_)  
te_df.head()
```

9. Preview dataframe:

	abrasive cleaner	artif. sweetener	baby cosmetics	baby food	bags	baking powder	bathroom cleaner	beef	berries	beverages	...	uht- milk	vinegar	waffles	whipped/sour cream	whisky	white bread	white wine	whole milk	yogurt	zwieback
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	True	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	True	False	False

5 rows × 169 columns

07

# Modelling

Pembuatan Model *Machine Learning*



# Modelling

Model dibentuk menggunakan algoritma apriori dengan mengimplementasikan library mlxtend.

```
frequent_itemsets = apriori(te_df, min_support=0.015,  
use_colnames=True, max_len=4)  
for i in frequent_itemsets['itemsets']:  
    for j in i:  
        j = str.lower(j)  
Frequent_itemsets
```

Preview dari rule yang terbentuk adalah sebagai berikut:

	support	itemsets
0	0.017692	(baking powder)
1	0.052466	(beef)
2	0.033249	(berries)
3	0.026029	(beverages)
4	0.080529	(bottled beer)
...	...	...
175	0.023183	(whole milk, root vegetables, other vegetables)
176	0.017082	(whole milk, other vegetables, tropical fruit)
177	0.022267	(yogurt, whole milk, other vegetables)
178	0.015557	(yogurt, whole milk, rolls/buns)
179	0.015150	(yogurt, whole milk, tropical fruit)

180 rows × 2 columns

08

# Evaluation

Pembuatan Model *Machine Learning*

# Evaluation

Melakukan pengujian dengan metrics confidence, dengan menggunakan library mlxtend.

```
association_rules_data = association_rules(frequent_itemsets,  
metric="confidence", min_threshold=0.015)  
association_rules_data
```

Preview dari matrix yang didapat :

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(other vegetables)	(beef)	0.193493	0.052466	0.019725	0.101944	1.943066	0.009574	1.055095
1	(beef)	(other vegetables)	0.052466	0.193493	0.019725	0.375969	1.943066	0.009574	1.292416
2	(root vegetables)	(beef)	0.108998	0.052466	0.017387	0.159515	3.040367	0.011668	1.127366
3	(beef)	(root vegetables)	0.052466	0.108998	0.017387	0.331395	3.040367	0.011668	1.332628
4	(whole milk)	(beef)	0.255516	0.052466	0.021251	0.083168	1.585180	0.007845	1.033487
...	...	...	...	...	...	...	...	...	...
233	(yogurt, tropical fruit)	(whole milk)	0.029283	0.255516	0.015150	0.517361	2.024770	0.007668	1.542528
234	(whole milk, tropical fruit)	(yogurt)	0.042298	0.139502	0.015150	0.358173	2.567516	0.009249	1.340701
235	(yogurt)	(whole milk, tropical fruit)	0.139502	0.042298	0.015150	0.108601	2.567516	0.009249	1.074380
236	(whole milk)	(yogurt, tropical fruit)	0.255516	0.029283	0.015150	0.059292	2.024770	0.007668	1.031900
237	(tropical fruit)	(yogurt, whole milk)	0.104931	0.056024	0.015150	0.144380	2.577089	0.009271	1.103265

# Evaluation

Untuk menggunakan model yang sudah dibuat, digunakan fungsi `find()` yang didefinisikan sebagai berikut:

```
def find(items, frequent_itemsets):
    out = []
    for i in range(len(frequent_itemsets)):
        if set(items).issubset(frequent_itemsets['itemsets'].iloc[i]):
            out.extend(frequent_itemsets['itemsets'].iloc[i])
    out = list(set([x for x in out if x not in items]))
    return out
```

Contoh penggunaan dan keluarannya adalah sebagai berikut:

```
find(['whole milk', 'root vegetables'], frequent_itemsets)
```

```
find(['whole milk', 'root vegetables'], frequent_itemsets)
['other vegetables']
```

# Evaluation

```
find(['whole milk'], frequent_itemsets)
```

```
find(['whole milk'], frequent_itemsets)
['root vegetables',
'sausage',
'citrus fruit',
'frankfurter',
'beef',
'rolls/buns',
'white bread',
'margarine',
'newspapers',
'other vegetables',
'pip fruit',
'butter',
'pastry',
'soda',
'chicken',
'sugar',
'whipped/sour cream',
'yogurt',
'shopping bags',
'brown bread',
'domestic eggs',
'bottled water',
'fruit/vegetable juice',
'chocolate',
'coffee',
'bottled beer',
'napkins',
'cream cheese',
'curd',
'pork',
'tropical fruit',
'frozen vegetables']
```

08

# Deployment

Menempatkan Model Pada Streamlit Cloud

# Deployment

Interface untuk sistem rekomendasi yang dideploy dibangun dengan menggunakan library streamlit. Library lain yang digunakan dalam deployment adalah library pickle untuk melakukan load model. Kode untuk import library adalah sebagai berikut:

```
import streamlit as st
import pickle
```

Sebelum membangun interface, model yang sudah dibuat ketika modelling harus diload terlebih dahulu.

```
file = open('model', 'rb')
data = pickle.load(file)
file.close()
```

# Deployment

Pada interface, di bagian atas dibuat judul program. Kemudian terdapat multiselect untuk memilih item produk dari daftar. Maksimal item yang dapat dipilih adalah dua karena rule set terpanjang pada model berukuran 3.

```
st.title('Sistem Rekomendasi')

opt = data[0]
symbols = st.multiselect("Pilih item yang akan dibeli (Maksimal 2 item): ", opt['item'], opt['item'][:2], max_selections=2)
```



# Deployment

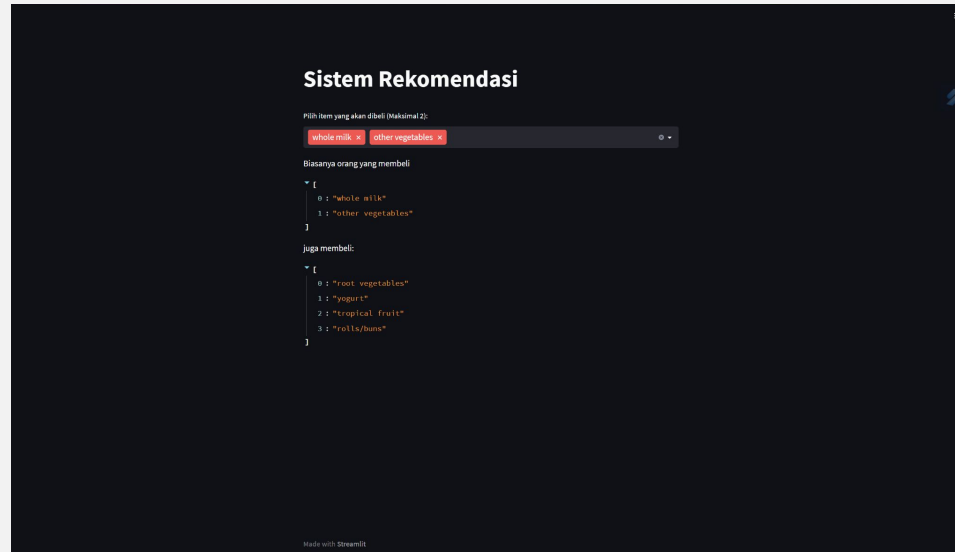
Masukan pada multiselect akan diproses dengan fungsi find seperti pada tahap evaluation. Kemudian hasilnya akan ditampilkan.

```
def find(items, frequent_itemsets):
    out = []
    for i in range(len(frequent_itemsets)):
        if set(items).issubset(frequent_itemsets['itemsets'].iloc[i]):
            out.extend(frequent_itemsets['itemsets'].iloc[i])
    out = list(set([x for x in out if x not in items]))
    return out

st.write('Biasanya orang yang membeli ', symbols, 'juga membeli: ')
st.write(find(symbols, data[1]))
```

# Deployment

Hasil deployment dapat dilihat pada link <https://feminovialina-simple-apriori-recommendation-system-d-app-93n7sm.streamlit.app/>. Preview hasil deployment adalah sebagai berikut:



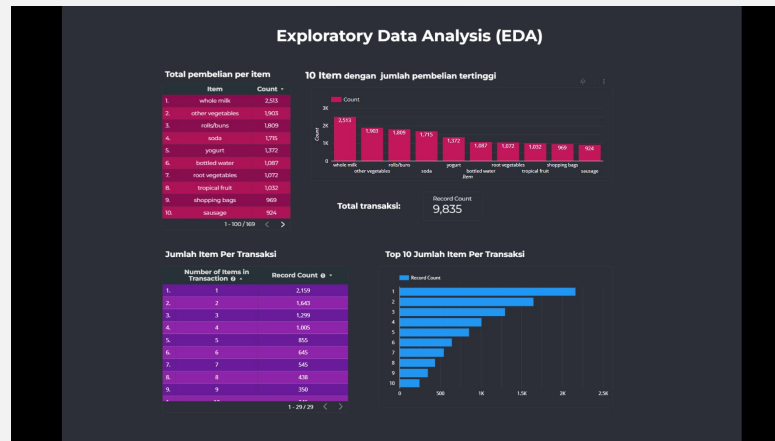
08

# Feedback

Mendapatkan Umpan Balik tentang Kinerja Model

# Feedback

Dashboard sebagai salah satu hasil akan memberikan visualisasi dan laporan dari hal-hal yang telah dilakukan pada tahap sebelumnya. Dashboard dibuat menggunakan Google Data Studio dan dapat diakses melalui <https://datastudio.google.com/reporting/5bcdc6f8-d0a5-42ae-b76d-91433869be18>.



# Thanks!

Do you have any questions?

