

CODIG COMPETITION 3.0

PROPOSAL PENAMBANGAN DATA (DATA MINING)



TIM CODEQUESTA

Efisiensi Market Basket Analysis

Dalam Pemaketan BULOG Pada masa Pandemi

DISUSUN OLEH:

ROBBY AUZIQNI 41518010085

NUR ISTIQAMAH ISKANDAR ALAM 41518010071

RIZQI FAUZAN 41518010080

UNIVERSITAS MERCU BUANA

JAKARTA BARAT

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Di Era yang serba digital saat ini, teknologi komputerisasi adalah hal yang sangat krusial dalam hampir segala lini kehidupan, baik dalam hal berbisnis maupun menentukan langkah dalam manage suatu permasalahan yang cukup kompleks sekalipun. Dalam dunia digital pula kita tidak asing juga dengan data, yang merupakan sekumpulan fakta yang tercantum, dimana data ini menggambarkan keadaan atau suatu kondisi dari seseorang, atau perangkat yang kita miliki. Dengan adanya revolusi industri 4.0 kita dapat melihat bahwa perkembangan teknologi sangatlah pesat, namun dibalik cepatnya perkembangan teknologi tersebut, kita juga menerima banyaknya sebaran data yang kita hasilkan dan dikumpulkan dari teknologi atau aplikasi yang telah kita ciptakan tersebut. Ini menciptakan adanya suatu fenomena yang kita sebut *Tsunami Data*.

Menanggapi fenomena tersebut, kita seharusnya dapat menjadikan fenomena tersebut menjadi suatu keuntungan bagi kita. Dari data yang telah kita miliki dalam aplikasi yang telah dibuat, kita dapat mencari pola ataupun pengetahuan dalam banyaknya data yang telah kita miliki dengan metode *Data Mining*

Penerapan Data mining adalah salah satu konsep yang digunakan pada masa kini untuk menemukan pengetahuan yang tersembunyi di dalam database atau di dalam dataset yang kita miliki, yang mana dataset tersebut sudah di ekstrak dari database kita. Data mining merupakan proses semi otomatis yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi pengetahuan potensial dan berguna yang tersimpan di dalam database besar. (Turban et al, 2005). Data mining adalah bagian dari proses KDD (Knowledge Discovery in Databases) yang terdiri dari beberapa tahapan seperti pemilihan data, pra pengolahan, transformasi, data mining, dan evaluasi hasil (Maimon dan Last, 2000). KDD secara umum juga dikenal sebagai pangkalan data.

Adapun kasus yang kita hadapi saat ini ialah banyaknya sebaran data yang telah masuk dalam riwayat transaksi bulog, pada tahun 2019. Data yang kita dapat, berisi riwayat transaksi selama satu tahun belakang. dari data yang kita peroleh kita dapat melihat, bahwa dari sebaran data tersebut kita dapat merumuskan, item apa saja yang paling sering dibeli berdasarkan frekuensi pembelian, yang mana pada akhirnya hasil item yang paling sering muncul tersebut, dapat kita olah menjadi paket (bundling) item, demi efisiensi pendistribusian pada masa pandemi COVID-19. Hal ini dapat dilakukan dengan *Association Rule Mining*, dalam terapan *Market Basket Analysis* menggunakan algoritma FP-Growth.

1.2. Rumusan Masalah

Adapun masalah yang akan dihadapi dapat dirumuskan sebagai berikut:

- Bagaimana cara menemukan komoditi apa saja yang dibeli secara bersamaan?
- Bagaimana cara memprediksi besar kemungkinan bahwa suatu komoditi akan dibeli jika konsumen membeli suatu komoditi tertentu?

1.3. Batasan Masalah

Dalam melakukan penambangan data pada kasus ini kita diberi batasan berupa:

- Data yang akan ditambang berasal dari riwayat transaksi BULOG pada tahun 2019
- Data tidak melihat kepada kuantitas barang yang dibeli tiap transaksi.
- Metode penambangan yang akan digunakan adalah association mining dengan menggunakan algoritma FP-growth

BAB 2 TUJUAN DAN MANFAAT

2.1. Tujuan

Tujuan dari penambangan data pada kasus ini adalah untuk memprediksikan serangkaian komoditi untuk dipaketkan berdasarkan jenis dengan tepat agar mempermudah penyediaan sembako kepada pedagang dan distributor sesuai dengan kebutuhan tiap pedagang dan distributor.

2.2. Manfaat

Di tengah masa pandemi Covid-19 saat ini, banyak bahan sembako yang mulai naik harganya secara perlahan, hal tersebut tentu sangat berpengaruh bagi masyarakat yang memiliki pemasukan menengah ke bawah.

Manfaat dari memprediksikan serangkaian sembako untuk dipaketkan adalah sebagai salah satu solusi untuk mencegah pemborosan sembako dan setiap pembelian sembako berbentuk paket akan diberikan potongan harga sehingga konsumen dapat terbantu dalam memenuhi kebutuhan pangannya. Selain itu, hal ini juga mempengaruhi efisiensi pengiriman.

BAB 3 METODE PENAMBANGAN DATA

3.1 Association Rule Mining

Association Rule Mining adalah prosedur yang bertujuan untuk mengamati pola, korelasi, atau asosiasi yang sering muncul dari kumpulan data. Sebuah Association Rule memiliki 2 bagian:

- Sebuah **Antecedent (if)**
- Sebuah **Consequent (then)**

Sebuah antecedent adalah sesuatu yang ditemukan di data, dan sebuah consequent adalah item yang ditemukan dalam kombinasi dengan antecedent. Dalam association rule mining, kita harus memperhatikan berbagai pengukuran yang tersedia untuk memahami kekuatan asosiasi antara keduanya. Di antara pengukuran tersebut ialah:

- **Support**

Ukuran ini memberikan gambaran tentang seberapa sering suatu itemset berada dalam semua transaksi. Support digunakan untuk mengukur kelimpahan atau frekuensi (sering diartikan sebagai signifikansi atau kepentingan) dari kumpulan item dalam database. Secara matematis, support adalah pecahan dari jumlah total transaksi di mana kumpulan item terjadi.

$$\text{Support}(\{X\} \rightarrow \{Y\}) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Total number of transactions}}$$

Jika kumpulan item kebetulan memiliki support yang sangat rendah, ini menunjukkan ketidakcukupan informasi tentang hubungan antara item-itemnya.

- **Confidence**

Ukuran ini adalah probabilitas untuk melihat konsekuensi dalam transaksi yang mengandung antecedent. Confidence merupakan indikasi seberapa sering sebuah aturan terbukti benar.

$$\text{Confidence}(\{X\} \rightarrow \{Y\}) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Transactions containing } X}$$

- **Lift**

Ukuran ini digunakan untuk mengukur seberapa sering antecedent dan konsekuensi dari sebuah aturan terjadi bersama-sama daripada yang kita harapkan jika mereka independen secara statistik

$$\rightarrow \{Y\}) = \frac{(\text{Transactions containing both } X \text{ and } Y) / (\text{Transactions containing } X)}{\text{Fraction of transactions containing } Y}$$

Jika sebuah aturan memiliki lift 1, itu akan menyiratkan bahwa probabilitas kemunculan antecedent dan konsekuensi dari itu tidak tergantung satu sama lain. Jika dua peristiwa tidak bergantung satu sama lain, tidak ada aturan yang dapat dibuat yang melibatkan kedua peristiwa tersebut.

Jika peningkatan > 1, itu memungkinkan kita mengetahui sejauh mana kedua kemunculan tersebut bergantung satu sama lain, dan membuat aturan tersebut berpotensi berguna untuk memprediksi konsekuensi dalam kumpulan data mendatang.

Jika lift < 1, itu memberi tahu kami bahwa barang-barang tersebut saling menggantikan. Artinya, keberadaan satu item berpengaruh negatif terhadap keberadaan item lainnya begitu pula sebaliknya.

Salah satu penerapan dari association rule adalah untuk Market Basket Analysis. Market Basket Analysis adalah salah satu teknik utama yang digunakan oleh retailer besar untuk mengungkap asosiasi antar item. Ini bekerja dengan mencari kombinasi item yang sering terjadi bersamaan dalam transaksi. Dengan kata lain, ini memungkinkan retailer untuk mengidentifikasi hubungan antara barang yang dibeli orang.

3.2. FP-Growth

Algoritma FP-Growth adalah metode yang efisien dan dapat diskalakan untuk menambang set lengkap pola frequent dengan pertumbuhan fragmen pola, menggunakan struktur prefix-tree yang diperluas untuk menyimpan informasi terkompresi dan penting tentang pola yang sering dinamai frequent-pattern tree (FP-tree).

Algoritma FP-Growth adalah cara alternatif untuk menemukan kumpulan item yang sering tanpa menggunakan generasi kandidat, sekaligus meningkatkan kinerja. Untuk banyak hal, ia menggunakan strategi "divide and conquer". Inti dari metode ini adalah penggunaan struktur data khusus bernama frequent-pattern tree (FP-tree), yang menyimpan informasi asosiasi itemset.

FP-tree adalah struktur data yang mewakili kumpulan data dalam bentuk pohon. Setiap transaksi dibaca dan kemudian dipetakan ke jalur di pohon FP. Ini dilakukan sampai semua transaksi telah dibaca. Transaksi berbeda yang memiliki subset umum memungkinkan pohon tetap compact karena jalurnya tumpang tindih.

Secara sederhana, algoritma FP-Growth bekerja sebagai berikut: pertama, algoritma tersebut memampatkan basis data dengan membuat FP-tree untuk merepresentasikan item yang sering. Kemudian, ia membagi database yang dikompresi menjadi satu set database bersyarat, masing-masing terkait dengan satu pola yang sering. Akhirnya, setiap database tersebut ditambang secara terpisah. Selengkapnya, algoritma FP-Growth adalah sebagai berikut:

Algorithm: FP_growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input:

- D , a transaction database;
- min_sup , the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:

- (a) Scan the transaction database D once. Collect F , the set of frequent items, and their support counts. Sort F in support count descending order as L , the list of frequent items.
- (b) Create the root of an FP-tree, and label it as "null." For each transaction $Trans$ in D do the following.
Select and sort the frequent items in $Trans$ according to the order of L . Let the sorted frequent item list in $Trans$ be $[p|P]$, where p is the first element and P is the remaining list. Call `insert_tree([p|P], T)`, which is performed as follows. If T has a child N such that $N.item_name = p.item_name$, then increment N 's count by 1; else create a new node N , and let its count be 1, its parent link be linked to T , and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call `insert_tree(P, N)` recursively.

2. The FP-tree is mined by calling `FP_growth(FP_tree, null)`, which is implemented as follows.

procedure `FP_growth(Tree, α)`

- (1) **if** $Tree$ contains a single path P **then**
- (2) **for each** combination (denoted as β) of the nodes in the path P
- (3) generate pattern $\beta \cup \alpha$ with *support_count* = *minimum support count of nodes in β* ;
- (4) **else for each** a_i in the header of $Tree$ {
- (5) generate pattern $\beta = a_i \cup \alpha$ with *support_count* = $a_i.support_count$;
- (6) construct β 's conditional pattern base and then β 's conditional FP_tree $Tree_\beta$;
- (7) **if** $Tree_\beta \neq \emptyset$ **then**
- (8) call `FP_growth(Tree β , β)`; }

FP-growth algorithm for discovering frequent itemsets without candidate generation.

BAB 4 HASIL DAN PEMBAHASAN

4.1 Desain

- Pengumpulan Data
- Preprocessing
 - Penyeleksian Data
 - Transformasi Data
- Penerapan Algoritma
- Evaluasi

4.2 Implementasi

4.2.1 Pengumpulan Data

Dataset yang dipakai dalam makalah ini adalah data transaksi BULOG selama 1 tahun (2019). Sampel dari data tersebut bisa dilihat pada *Tabel 4.2.1.1*

	Tanggal SPS	Tanggal Bayar	No. SPS	Saluran	ID Pembeli	KAB/ Kota	Nama Komoditi	Jenis Komoditi	Merk Komoditi	Kemasan	Satuan	Kolli Terjual	Kuantum Terjual (Kg/L)
0	01-02-2019	01-02-2019	02130/01/2019/09001/SPST	KOPERASI/ASOSIASI	77844	KOTA ADM. JAKARTA SELATAN	DAGING KERBAU	FOREQUARTER	DAGING KITA	18	Kg	1	18
1	01-02-2019	01-02-2019	02010/01/2019/09001/SPST	TOKO/AGEN/PENGECEK	72580	KOTA ADM. JAKARTA TIMUR	DAGING KERBAU	FOREQUARTER	DAGING KITA	20	Kg	5	100
2	17-12-2018	01-02-2019	01200/12/2018/09001/SPST	HOREKA	55107	KAB. TANGERANG	DAGING KERBAU	FOREQUARTER	DAGING KITA	20	Kg	12	240
3	01-02-2019	01-02-2019	02007/01/2019/09001/SPST	HOREKA	55107	KAB. TANGERANG	DAGING KERBAU	FOREQUARTER	DAGING KITA	20	Kg	7	140
4	01-03-2019	01-02-2019	00007/01/2019/09001/SPST	RPK-NON BPNT/Program Sembako	67725	KOTA ADM. JAKARTA PUSAT	DAGING KERBAU	HINDQUARTER	DAGING KITA	1	Kg	75	75

Table 4.2.1.1

4.2.2 Preprocessing

Dataset yang didapatkan merupakan data mentah. Untuk itu diperlukan proses preprocessing. Hal ini bertujuan untuk mengubah data ke dalam format yang memungkinkan adanya teknik data mining. Di dalam makalah ini, proses preprocessing dibagi menjadi dua, yakni penyeleksian data dan transformasi data.

4.2.2.1 Penyeleksian Data

Dataset yang dipakai memiliki banyak atribut, namun kami menilai harusnya untuk menyeleksi kolom yang dianggap penting saja. Dalam hal ini, kami hanya memilih

kolom “No. SPS”, “Nama Komoditi”, “Jenis Komoditi”, dan “Merk Komoditi”. Hal ini dilakukan karena tujuan dari penambahan data ini adalah untuk mengetahui item atau

	No. SPS	Nama Komoditi	Jenis Komoditi	Merk Komoditi
13	00001/01/2019/09001/SPST	DAGING KERBAU	HINDQUARTER	DAGING KITA
14	00001/01/2019/09001/SPST	MINYAK GORENG	MINYAK GORENG KITA	MINYAK GORENG KITA
15	00001/01/2019/09020/SPST	BERAS	THAILAND 5%	BERAS POLOS
12	00002/01/2019/09001/SPST	DAGING KERBAU	HINDQUARTER	DAGING KITA
11	00003/01/2019/09001/SPST	DAGING KERBAU	HINDQUARTER	DAGING KITA

Table 4.2.2.1.1

	No. SPS	Komoditi
13	00001/01/2019/09001/SPST	DAGING KERBAU - HINDQUARTER (DAGING KITA)
14	00001/01/2019/09001/SPST	MINYAK GORENG - MINYAK GORENG KITA (MINYAK GOR...
15	00001/01/2019/09020/SPST	BERAS - THAILAND 5% (BERAS POLOS)
12	00002/01/2019/09001/SPST	DAGING KERBAU - HINDQUARTER (DAGING KITA)
11	00003/01/2019/09001/SPST	DAGING KERBAU - HINDQUARTER (DAGING KITA)

Table 4.2.2.1.2

komoditi mana saja yang kemungkinan besar dibeli / dipesan bersamaan dalam satu transaksi. Hasil nya bisa dilihat seperti pada *Tabel 4.2.2.1.1*

	TID	Item
13	T0001	I1
14	T0001	I2
15	T0002	I3
12	T0003	I1
11	T0004	I1
...
9800	T5704	I7
9798	T5705	I7
9799	T5706	I7
9797	T5707	I7
9796	T5708	I7

Table 4.2.2.1.4

Setelah itu, pada *Tabel 4.2.2.1.2*, kami mencoba untuk menyederhanakan kolom, menggabung kolom “Nama Komoditi”, “Jenis Komoditi”, dan “Merk Komoditi” menjadi satu, agar memudahkan dalam pemrosesan. Setelah proses penyeleksian kolom selesai, bisa dilihat bahwa dari sisi penamaan nilai, data dirasa masih sangat rumit, maka kami perlu membuat data tersebut lebih sederhana lagi. Hal ini dapat dilakukan dengan memberikan label pada setiap item dan transaksi. Langkah selanjutnya adalah mengubah nilai item dan transaksi menjadi sesuai dengan labelnya. Lihat *Tabel 4.2.2.1.3 & Tabel 4.2.2.1.4*.

Kode		Komoditi
0	I1	DAGING KERBAU - HINDQUARTER - DAGING KITA
1	I2	MINYAK GORENG - MINYAK GORENG KITA - MINYAK GO...
2	I3	BERAS - THAILAND 5% - BERAS POLOS
3	I4	BERAS - KHUSUS - BERAS PANDAN WANGI 5KG
4	I5	BERAS - PREMIUM 10% - BERAS KITA PREMIUM
5	I6	GULA - GKP EKS RAW SUGAR PT.JMR - MANIS KITA
6	I7	DAGING KERBAU - FOREQUARTER - DAGING KITA
7	I8	TEPUNG TERIGU - TERIGU KITA - TERIGU KITA
8	I9	BERAS - VIETNAM 5% - BERAS POLOS
9	I10	BERAS - KHUSUS - BERASKITA BERAS HITAM
10	I11	BERAS - PREMIUM 15% - BERAS KITA PREMIUM

Table 4.2.2.1.3

4.2.2.2 Transformasi Data

Untuk menyesuaikan dengan algoritma yang dipakai, data harus ditransformasi sesuai format terlebih dahulu. Algoritma FP-Growth membutuhkan data itemset. Dalam hal ini, format data seperti pada *Tabel 4.2.2.1.4* dirasa belum cukup untuk memenuhi kriteria, karena data yang dibutuhkan adalah data item yang digrup berdasarkan jumlah transaksi. Untuk itulah data ditransformasi seperti pada *Tabel 4.2.2.2.1*.

	TID	Item
0	T0001	I1,I2
1	T0002	I3
2	T0003	I1
3	T0004	I1
4	T0005	I4,I2
...
5703	T5704	I7
5704	T5705	I7
5705	T5706	I7
5706	T5707	I7
5707	T5708	I7

Table 4.2.2.2.1

4.2.3 Penerapan Algoritma

Tahap selanjutnya adalah penerapan algoritma FP-Growth dari data yang sudah diformat. Kami menetapkan minimal support sebesar 30%, maka di dapatkanlah hasil pada *Tabel 4.2.3.1*. Dengan hasil ini, maka kami mendapati association rule yang bisa dibangun pada *Tabel 4.2.3.2*.

	support	itemsets
0	0.721172	(I2)
1	0.525520	(I6)
2	0.368620	(I5)
3	0.433837	(I8)
4	0.439509	(I11)
5	0.403592	(I6, I2)
6	0.346881	(I8, I2)
7	0.315690	(I11, I2)

Table 4.2.3.1

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(I6)	(I2)	0.525520	0.721172	0.403592	0.767986	1.064913	0.024601	1.201770
1	(I8)	(I2)	0.433837	0.721172	0.346881	0.799564	1.108701	0.034009	1.391109
2	(I11)	(I2)	0.439509	0.721172	0.315690	0.718280	0.995989	-0.001271	0.989733

Table 4.2.3.2

4.4 Analisis

Dengan melihat pada Tabel 4.2.3.1 dan association rule yang terbangun pada Tabel 4.2.3.2, kami mendapati rata - rata itemset memiliki nilai **Confidence** sekitar 70%. Kami dapat menarik asumsi bahwa ketika konsumen membeli **I6:”GULA - GKP EKS RAW SUGAR PT.JMR (MANIS KITA)”**, **I8:”TEPUNG TERIGU - TERIGU KITA (TERIGU KITA)”**, ataupun **I11:”BERAS - PREMIUM 15% (BERAS KITA PREMIUM)”**, 70% kemungkinan **I2:”MINYAK GORENG - MINYAK GORENG KITA (MINYAK GORENG KITA)”** akan dibeli bersamaan. Nilai **lift** yang didapatkan menunjukkan bahwa kemungkinan - kemungkinan tersebut bukan sebuah kemungkinan acak. Dengan itu, item - item tersebut dapat dipaketkan bersamaan untuk efisiensi pengiriman dalam masa pandemi saat ini. Namun demikian, jika dilihat dari nilai **support** yang kecil, tingkat kredibilitas asumsi ini masih rendah. Hal ini kemungkinan besar karena kurangnya data yang dimiliki.

BAB 5 KESIMPULAN

Kami telah melakukan kegiatan penambangan data (Data Mining) dengan menggunakan teknik Association Rule Mining menggunakan algoritma FP-Growth. Kami melakukan analisa berdasarkan jumlah item yang bersamaan pada tiap transaksi selama setahun, untuk mengetahui tingkat besarnya kemungkinan item untuk dipesan bersamaan. Hal ini bertujuan untuk meningkatkan efektifitas dan efisiensi pendistribusian barang terutama dalam kondisi Pandemi COVID-19 saat ini.

Efektifitas dan efisiensi pendistribusian dapat diraih diantaranya dengan cara pemaketan barang - barang yang kemungkinan besar dipesan secara bersamaan. Hasil yang didapat dari kegiatan ini menunjukkan tingkat confidence serta lift yang tinggi, namun tidak dapat dipungkiri hasil itu diraih dengan support yang rendah. Hal ini disebabkan karena item di dalam transaksi sangat beragam, dalam artian data sangat bervariasi. Hal inilah yang lantas menyebabkan support pada tiap item rendah. Hasil yang didapat dapat dikembangkan dengan cara penyediaan data yang lebih banyak, ataupun peninjauan data secara lebih mendalam.

BAB 6 LAMPIRAN

6.1 Dokumentasi Tahapan Penambangan Data

6.1.1 Pengumpulan Data

▾ Initialize dataframe

```
[3] data = pd.read_csv('/content/drive/My Drive/dataset_transaksi_penjualan.csv')
data.head()
```

id	No. SPS	Saluran	ID Pembeli	KAB/ Kota	Nama Komoditi	Jenis Komoditi	Merk Komoditi	Kemasan	Satuan	Kolli Terjual	Kuantum Terjual (Kg/L)	Harga per Kemasan	Nominal Penjualan
2-19	02130/01/2019/09001/SPST	KOPERASI/ASOSIASI	77844	KOTA ADM. JAKARTA SELATAN	DAGING KERBAU	FOREQUARTER	DAGING KITA	18	Kg	1	18	1098000	1098000
2-19	02010/01/2019/09001/SPST	TOKO/AGEN/PENGECEK	72580	KOTA ADM. JAKARTA TIMUR	DAGING KERBAU	FOREQUARTER	DAGING KITA	20	Kg	5	100	1440000	7200000
2-19	01200/12/2018/09001/SPST	HOREKA	55107	KAB. TANGERANG	DAGING KERBAU	FOREQUARTER	DAGING KITA	20	Kg	12	240	1380000	16560000
2-19	02007/01/2019/09001/SPST	HOREKA	55107	KAB. TANGERANG	DAGING KERBAU	FOREQUARTER	DAGING KITA	20	Kg	7	140	1380000	9560000
2-19	00007/01/2019/09001/SPST	RPK-NON BPNT/Program Sembako	67725	KOTA ADM. JAKARTA PUSAT	DAGING KERBAU	HINDQUARTER	DAGING KITA	1	Kg	75	75	70000	5250000

6.1.2 Preprocessing

```
# Select Column
data = data.sort_values(by=['Tanggal Bayar', 'No. SPS'])[['No. SPS', 'Nama Komoditi', 'Jenis Komoditi', 'Merk Komoditi']]
data.head()

data["Komoditi"] = data["Nama Komoditi"] + " - " + data["Jenis Komoditi"] + " (" + data["Merk Komoditi"] + ")"
data = data.drop(['Nama Komoditi', 'Jenis Komoditi', 'Merk Komoditi'], axis=1)
data.head()
```

	No. SPS	Komoditi
13	00001/01/2019/09001/SPST	DAGING KERBAU - HINDQUARTER (DAGING KITA)
14	00001/01/2019/09001/SPST	MINYAK GORENG - MINYAK GORENG KITA (MINYAK GOR...
15	00001/01/2019/09020/SPST	BERAS - THAILAND 5% (BERAS POLOS)
12	00002/01/2019/09001/SPST	DAGING KERBAU - HINDQUARTER (DAGING KITA)
11	00003/01/2019/09001/SPST	DAGING KERBAU - HINDQUARTER (DAGING KITA)

```
[6] # Replace komoditi values
komoditi_list = data['Komoditi'].unique()
komoditi_list

replace_list = []
for i in range(len(komoditi_list)):
    replace_list.append('I'+str(i + 1))
replace_list

# Create legend table
def data = {
    'Kode': replace_list,
    'Komoditi': komoditi_list,
}

# Replace
data = data.replace(komoditi_list, replace_list)
data

# Replace SPS values as Transaction
sps_list = data['No. SPS'].unique()
sps_list
```

TID Item

13	T0001	I1
14	T0001	I2
15	T0002	I3
12	T0003	I1
11	T0004	I1
...
9800	T5704	I7
9798	T5705	I7
9799	T5706	I7
9797	T5707	I7

TID Item

0	T0001	I1,I2
1	T0002	I3
2	T0003	I1
3	T0004	I1
4	T0005	I4,I2
...
5703	T5704	I7
5704	T5705	I7
5705	T5706	I7
5706	T5707	I7
5707	T5708	I7

```
[8]: te = TransactionEncoder()
data_preprocess['Item'].tolist()
print(data_preprocess['Item'].tolist())
print(len(data_preprocess['Item'].tolist()))

# Change to suitable format for the algorithm
new_list = []
for i in data_preprocess['Item'].tolist():
    if len(i.split(',')) > 2:
        new_list.append(i.split(','))
print(new_list)
print(len(new_list))

te_ary = te.fit(new_list).transform(new_list)
df = pd.DataFrame(te_ary, columns=te.columns_)
df

[11,12', '13', '11', '11', '14,12', '15,16,12', '15', '11', '17', '17', '17', '17', '11,18', '11', '11', '16', '19', '15,14,15', '15', '110,111', '111', '15,12', '16', '12', '15', '14', '15', '113', '15', '16', '12', '18', '15', '14', '11', '111', '15', '16', '12', '11', '16', '12', '15', '16', '12'
1058
```

	I1	I10	I11	I12	I13	I14	I15	I17	I18	I19	I2	I20	I21	I22	I23	I24	I25	I26	I27	I28	I29	I30	I31	I32	I33
0	False	False	False	False	False	False	False	False	False	False	True	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	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	True	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	True	False	True	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	True	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
1053	False	False	True	False	False	False	False	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1054	False	False	True	False	False	False	False	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1055	False	False	True	False	False	False	False	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1056	False	False	False	False	False	False	False	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1057	False	False	True	False	False	True	True	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False

1058 rows x 38 columns

```
[9] f_itemsets = fpgrowth(df, min_support=0.3, use_colnames=True)
f_itemsets
```

	support	itemsets
0	0.721172	{I2}
1	0.525520	{I6}
2	0.368620	{I5}
3	0.433837	{I8}
4	0.439509	{I11}
5	0.403592	{I6, I2}
6	0.346881	{I8, I2}
7	0.315690	{I11, I2}

6.1.4 Evaluasi

▼ Create Association Rule

```
[ ] association_rules(f_itemsets, metric="confidence", min_threshold=0.7)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	{I6}	{I2}	0.721172	0.403592	0.767986	1.064913	0.024601	1.201770	
1	{I8}	{I2}	0.433837	0.721172	0.346881	0.799564	1.108701	0.034009	1.391109
2	{I11}	{I2}	0.439509	0.721172	0.315690	0.718280	0.995989	-0.001271	0.989733

pd.DataFrame(defdata, columns=['Kode', 'Komoditi'])

	Kode	Komoditi
0	I1	DAGING KERBAU - HINDQUARTER (DAGING KITA)
1	I2	MINYAK GORENG - MINYAK GORENG KITA (MINYAK GOR...
2	I3	BERAS - THAILAND 5% (BERAS POLOS)
3	I4	BERAS - KHUSUS (BERAS PANDAN WANGI SKG)
4	I5	BERAS - PREMIUM 10% (BERAS KITA PREMIUM)
5	I6	GULA - GKP EKS RAWUGAR PT.JMR (MANIS KITA)
6	I7	DAGING KERBAU - FOREQUARTER (DAGING KITA)
7	I8	TEPUNG TERIGU - TERIGU KITA (TERIGU KITA)
8	I9	BERAS - VIETNAM 5% (BERAS POLOS)
9	I10	BERAS - KHUSUS (BERASKITA BERAS HITAM)
10	I11	BERAS - PREMIUM 15% (BERAS KITA PREMIUM)
11	I12	GULA - GKP EKS RAWUGAR PT.JMR (GULA POLOS)
12	I13	BERAS - KHUSUS (BERAS BASMATI)
13	I14	BERAS - KHUSUS (BERAS MILKY)
14	I15	BERAS - KHUSUS (BERASKITA BERAS MERAH)