

IMPLEMENTASI ALGORITMA APRIORI












Hendri Kurniawan Prakosa

19/448706/PPA/05789

1. Prepare dataset

a. Download

<https://github.com/graphql-compose/graphql-compose-examples/tree/master/examples/northwind/data/csv>

 categories.csv	Preparation for big example: added sample data from Northwind.
 customers.csv	Preparation for big example: added sample data from Northwind.
 employee_territories.csv	Preparation for big example: added sample data from Northwind.
 employees.csv	Preparation for big example: added sample data from Northwind.
 order_details.csv	Preparation for big example: added sample data from Northwind.
 orders.csv	Preparation for big example: added sample data from Northwind.
 products.csv	Preparation for big example: added sample data from Northwind.
 regions.csv	Preparation for big example: added sample data from Northwind.
 shippers.csv	Preparation for big example: added sample data from Northwind.
 suppliers.csv	Preparation for big example: added sample data from Northwind.
 territories.csv	Preparation for big example: added sample data from Northwind.

Gambar 1 List data

Karena yang akan dianalisis adalah data transaksi maka data yang akan digunakan adalah

- data order detail (order_detail.csv) adalah data yang berisi list produk yang telah terjual pada tiap order-nya, berapa *quantity*-nya
- data order (order.csv) adalah data yang memuat informasi siapa customernya, kapan tanggal ordernya
- data kategori (categories.csv). data yang berisi kategori dari produk
- data produk (product.csv) adalah data yang berisi informasi tentang produk, seperti nama, kemasan dll.

Data tersebut masih perlu diolah lagi karena belum sesuai dengan format. Dapat dilihat pada gambar 2.

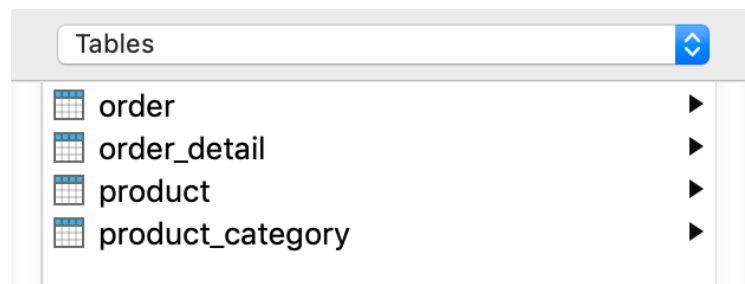
	A	B	C	D	E
1	orderID	productID	unitPrice	quantity	discount
2	10248	11	14.00	12	0
3	10248	42	0,43055556	10	0
4	10248	72	34.80	5	0
5	10249	14	0,79166667	9	0
6	10249	51	42.40.00	40	0
7	10250	41	0,34027778	10	0
8	10250	51	42.40.00	35	00.15
9	10250	65	0,72222222	15	00.15
10	10251	22	0,72222222	6	00.05
11	10251	57	0,66666667	15	00.05
12	10251	65	0,72222222	20	0

Gambar 2 Tampilan data order_detail.csv

Data order detail memperlihatkan bahwa pada kolom productID data yang tampil berupa id atau kode dari produk dan terdapat perulangan pada orderID yang sebenarnya memiliki arti bahwa dalam 1 orderID terdapat lebih dari 1 produk yang dibeli. Sementara informasi nama produk berada pada file yang terpisah (product.csv). Hal ini akan menyulitkan ketika akan diolah. Sehingga perlu digunakan query dalam database untuk membentuk data sesuai format yang diharapkan.

b. Import ke DBMS

Data direstore ke dalam database mysql dan menghasilkan 4 tabel, sebagai berikut (lihat gambar 3):



Gambar 3 List tabel transaksi

Dengan menggunakan query pada gambar 4.

```

1 SELECT order_detail.order_id, group_concat(product_name) AS product_name, `order`.`order_date`
2 FROM order_detail
3 JOIN product ON product.product_id = order_detail.product_id
4 JOIN `order` ON order.order_id = order_detail.order_id
5 GROUP BY `order_detail`.`order_id`

```

Gambar 4 Query data

Kemudian di ekspor dalam bentuk format *.csv sehingga mendapatkan data yang siap diolah seperti pada gambar 5.

	A	B	C
1	10248	Queso Cabrales,Singaporean Hokkien Fried Mee,Mozzarella di Giovanni	04/07/96
2	10249	Manjimup Dried Apples,Tofu	05/07/96
3	10250	Jack's New England Clam Chowder,Manjimup Dried Apples,Louisiana Fiery Hot Pepper Sauce	08/07/96
4	10251	Gustaf's KnvSckebrvdd,Ravioli Angelo,Louisiana Fiery Hot Pepper Sauce	08/07/96
5	10252	Geitost,Camembert Pierrot,Sir Rodney's Marmalade	09/07/96
6	10253	Gorgonzola Telino,Chartreuse verte,Maxilaku	10/07/96
7	10254	Guaranv° Fantv°stica,PVctv© chinois,Longlife Tofu	11/07/96
8	10255	Chang,Pavlova,Inlagd Sill,Raclette Courdavault	12/07/96
9	10256	Perth Pasties,Original Frankfurter grv°ne Sovüe	15/07/96
10	10257	Original Frankfurter grv°ne Sovüe,Schoggi Schokolade,Chartreuse verte	16/07/96
11	10258	Chang,Chef Anton's Gumbo Mix,Mascarpone Fabioli	17/07/96
12	10259	Sir Rodney's Scones,Gravad lax	18/07/96
13	10260	Tarte au sucre,Outback Lager,Jack's New England Clam Chowder,Ravioli Angelo	19/07/96
14	10261	Sir Rodney's Scones,Steeleye Stout	19/07/96
15	10262	Chef Anton's Gumbo Mix,Uncle Bob's Organic Dried Pears,Gnocchi di nonna Alice	22/07/96

Gambar 5 Data dalam format csv

2. Implementasi

a. Minimum support

Support adalah nilai popularitas suatu item yang didapat dari jumlah transaksi yang mengandung item tersebut dibagi jumlah semua transaksi yang ada, sehingga dapat dirumuskan sebagai berikut:

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Jumlah Transaksi}}$$

Dengan menggunakan query pada gambar 6

```
3 SELECT product.product_name, COUNT(order_detail.product_id) AS qty,  
4 (COUNT(order_detail.product_id)/830) AS support  
5 FROM order_detail  
6 JOIN product ON product.product_id = order_detail.product_id  
7 GROUP BY order_detail.product_id  
8 ORDER BY support
```

Gambar 6 Query mencari support minimal

Didapatkan list support tiap barang yang ditunjukkan oleh gambar 7. Nilai 830 pada query tersebut adalah jumlah transaksi keseluruhan yang ada pada data order.

	product_name	qty	support
1	Mishi Kobe Niku	5	0.0060
2	Genen Shouyu	6	0.0072
3	Chocolate	6	0.0072
4	Gravad lax	6	0.0072
5	Louisiana Hot Spiced Okra	8	0.0096
6	Schoggi Schokolade	9	0.0108
7	Valkoinen suklaa	10	0.0120
8	Chef Anton's Gumbo Mix	10	0.0120
9	Laughing Lumberjack Lager	10	0.0120
10	Grandma's Boysenberry Spread	12	0.0145

Gambar 7 Hasil query mencari support yang minimal

Sehingga dari hasil pada gambar 7 didapatkan **minimal support 0.006**

b. Confidence

Nilai confidence adalah nilai kemungkinan sebuah item B dibeli juga jika item A dibeli. Nilai confidence dapat dihitung dengan:

$$\text{Confidence} = P(B|A) = \frac{\Sigma \text{Transaksi mengandung A dan B}}{\Sigma \text{Transaksi mengandung A}}$$

Jika nilai confidence yang ditentukan mencapai 100% maka artinya adalah sudah pasti peluang item B akan ikut dibeli juga jika item A terbeli. Hal ini bergantung pada *behavior* dari data transaksi yang dipunya. Jika terlalu tinggi dalam menentukan nilai confidence maka rule yang diinginkan bisa jadi tidak terbentuk. Sehingga perlu ditetapkan batas bawah yang paling memungkinkan.

Karena penentuan nilai confidence sifatnya adalah *trial and error* maka pada kasus ini ditentukan **nilai confidence-nya mulai dari nilai 15%**.

c. Lift

Nilai lift adalah rasio terjualnya item B ketika A terbeli. Secara matematis dapat dihitung:

$$\text{Lift}(A \rightarrow B) = (\text{Confidence}(A \rightarrow B)) / (\text{Support}(B))$$

Jika nilai lift > 1 maka kemungkinan terbelinya kedua produk (A dan B) tersebut makin tinggi.

Sehingga berdasarkan pada hal tersebut, untuk mengetahui rule yang akan kita bangun memiliki kemungkinan yang tinggi, maka **nilai lift ditetapkan 2**

d. Implementasi dan hasil (kode lengkap terlampir)

Implementasi menggunakan Jupyter Notebook library apyori dari python untuk memproses data. Gambar 8 menunjukkan proses training pembentukan rule pada fungsi apriori yang ada pada library apyori.

```
In [6]: # training apriori
min_lift_dt = 2
support_dt = 0.006
conf_rule = 0.15
asc_rules = apriori(rows, min_support = support_dt, min_confidence = conf_rule, min_lift = min_lift_dt,
asc_rules_result = list(asc_rules)
```

Gambar 8

...

Hasilnya terlihat pada gambar 9.

```
In [10]: listRules
```

```
Out[10]: [['Alice Mutton', 'Geitost', {'conf': 0.15625}, {'lift': 3.5050675675675675}],  
          ['Louisiana Fiery Hot Pepper Sauce',  
           'Gnocchi di nonna Alice',  
           {'conf': 0.15625},  
           {'lift': 2.59375}],  
          ['Gorgonzola Telino',  
           'Mozzarella di Giovanni',  
           {'conf': 0.15789473684210528},  
           {'lift': 2.5696594427244586}],  
          ['Pavlova',  
           'Gorgonzola Telino',  
           {'conf': 0.1627906976744186},  
           {'lift': 2.6493388052895575}],  
          ['Original Frankfurter grüne Soße',  
           'Ikura',  
           {'conf': 0.15151515151515152},  
           {'lift': 3.309409888357257}],  
          ['Tourtière',  
           'Nord-Ost Matjeshering',  
           {'conf': 0.1875},  
           {'lift': 4.322916666666667}],  
          ['Tarte au sucre',  
           'Pâté chinois',  
           {'conf': 0.15151515151515152},  
           {'lift': 2.619949494949495}],  
          ['Sirop d'érable',  
           'Sir Rodney's Scones',  
           {'conf': 0.20512820512820515},  
           {'lift': 7.094017094017095}]]
```

Gambar 9

3. Analisis hasil

Berdasarkan beberapa kali percobaan dengan menggunakan nilai confidence 15%, nilai lift sebanyak 2 dan minimum support yang berbeda-beda, didapatkan rule yang paling optimal adalah ketika nilai minimum supportnya 0.002. Terlihat pada tabel 1.

Tabel 1 Perbandingan nilai minimum dengan jumlah rules dan waktu

Nilai Minimum Support	Jumlah Rules	Status Proses
0.006	8	Finished
0.005	8	Finished
0.004	13	Finished
0.003	38	Finished
0.002	118	Finished
0.001	-	Unfinished

Dengan menggunakan optimal minimum support yang bernilai 0.002, percobaan dilakukan kembali dengan menaikkan nilai confidence hingga 100% dan menghasilkan rule sebanyak 35 rules. Salah satu rulenya adalah terlihat pada gambar 10

```
In [31]: listRules[0]
Out[31]: ['Gnocchi di nonna Alice',
          'Alice Mutton',
          'Raclette Courdavault',
          {'conf': 1.0},
          {'lift': 16.6}]
```

Gambar 10 Contoh rule yang dihasilkan dengan nilai confidence 100%

Berdasarkan rule pada gambar 10 dapat diartikan bahwa customer yang membeli produk 'Gnocchi di nonna Alice' dan 'Alice Mutton' maka pasti akan membeli juga produk 'Raclette Courdavault'.