

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Market Basket Analysis

*Market basket analysis* adalah suatu proses analisa perilaku pembelian kosumen dengan menemukan asosiasi antar item yang berbeda yang kosumen tempatkan pada keranjang belanjaan mereka. Asosiasi yang ditemukan dapat membantu retailer mengembangkan strategi pemasaran dengan mengetahui barang mana saja yang seringkali dibeli secara bersamaan oleh kosumen. Misalnya, jika kosumen membeli susu, berapa kemungkinan mereka juga membeli roti (dan roti macam apa) dalam satu transaksi. Informasi ini dapat menyebabkan peningkatan penjualan dengan melakukan pemasaran selektif dan perencanaan tata ruang dan letak (Han, Kamber, & Pei, 2012). Pihak retailer dapat menggunakan informasi ini untuk menempatkan barang-barang yang sering terjual bersama di area yang sama, sedangkan situs ecommerce juga dapat menggunakannya untuk menentukan layout penyajian barang yang ditampilkan di situs mereka.

#### 2.2 Association Rules Mining

*Association Rules Mining* adalah suatu prosedur untuk mencari hubungan antar item dalam suatu dataset. Association rules meliputi dua tahap (Ulmer, 2002), antara lain sebagai berikut.

1. Mencari kombinasi yang paling sering terjadi dari suatu itemset.
2. Mendefinisikan *condition* dan *result* (untuk *conditional association rules*).

Dalam menentukan suatu *association rules*, terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran dalam *association rules* sebagai berikut.

1. *Support*

*Support* merupakan suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu item/itemset dari keseluruhan transaksi. Ukuran ini

menentukan apakah suatu item/itemset layak untuk dicari confidence-nya. Misalnya, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa item A dan B dibeli secara bersamaan).

## 2. *Confidence*

Confidence merupakan suatu ukuran yang menunjukkan hubungan antar dua item secara conditional. Misalnya, seberapa sering item B dibeli jika orang membeli item A.

Kedua ukuran ini berguna dalam menentukan *interesting association rules*, yaitu untuk dibandingkan dengan batasan (*threshold*) yang ditentukan oleh peneliti. Batasan tersebut umumnya terdiri dari minimal *support* dan minimal *confidence* (Budhi, Lim, & Prayitno, 2005). Minimum *support* adalah parameter yang digunakan sebagai batasan frekuensi kejadian atau *support count* yang harus dipenuhi suatu kelompok data untuk dapat dijadikan aturan. Sedangkan minimum *confidence* adalah parameter yang mendefinisikan minimum level dari *confidence* yang harus dipenuhi oleh aturan yang berkualitas (Yuniati, 2016). Minimum *support* dan minimum *confidence* biasanya ditentukan sendiri oleh peneliti. Semakin tinggi minimum *support* dan minimum *confidence* yang ditentukan, maka semakin sedikit *rules* yang dihasilkan karena banyak yang terseleksi. Jika  $support \geq \text{minimum support}$  dan  $confidence \geq \text{minimum confidence}$ , maka *rule* tersebut bisa dikatakan sebagai *interesting rule*. *Association rules* biasanya dinyatakan dalam bentuk: {roti, mentega} → {susu} (support = 40%, confidence = 50%), yang berarti 50% dari transaksi di *database* yang memuat item roti dan mentega juga memuat item susu. Sedangkan 40% dari seluruh transaksi yang ada di *database* memuat ketiga item itu (Nataliani, Beeh, & Dewi, 2010).

### 2.3 Association Rules Networks

*Association Rules* pada algoritma konvensional tidak dapat memberikan saran yang efisien terhadap hubungan antar barang secara keseluruhan pada suatu *database* transaksi yang besar. *Association Rules Networks* diusulkan pertama kali oleh T. Raeder dan N. V. Chawla pada tahun 2010 untuk mengatasi permasalahan

tersebut. ARN memvisualisasikan pola pembelian barang pada database transaksi dalam bentuk *network* yang dapat menggambarkan hubungan langsung dan tidak langsung suatu barang dengan barang lainnya. Pada ARN, *rules* yang ditemukan oleh algoritma *association rules mining* dapat dianalisis, dipangkas, dan diintegrasikan ke dalam konteks objek yang spesifik. Jika terdapat item yang menarik, maka kita dapat membentuk suatu *network* yang terdiri dari *item-item* yang berhubungan dan menggunakan *network* tersebut untuk membentuk suatu model yang dapat diperiksa dengan metode lainnya.

Beberapa fitur yang dimiliki ARN adalah sebagai berikut.

1. *Pruning in context*; ARN dapat digunakan untuk memangkas *rules* untuk menjaga objek utama
2. *Network structure*; *network* ini digunakan untuk menggambarkan hubungan antara *item*
3. *Hypothesis generation for evaluation*; ARN dapat menjembatani *output* yang dibangkitkan oleh model *Association Rule* dengan analisis untuk evaluasi.

### 2.3.1 Network of Products

*Networks* menggambarkan struktur model yang membantu dalam menganalisis dan memahami bagaimana suatu skema yang berbeda terjadi secara bersamaan. *Network of products* merupakan suatu model yang terdiri dari *node* dan *edge*. *Node* merupakan titik yang merepresentasikan suatu barang atau *item*, sedangkan *edge* merupakan garis yang menghubungkan suatu titik dengan titik lainnya. Jika terdapat sebuah *edge* yang di antara dua *node*, maka dapat diartikan bahwa kedua barang tersebut dibeli bersama dalam sekali atau lebih transaksi (Verma, 2017).



## Gambar 2.2 Ilustrasi Communities of Products

### 2.3.3 Modularity

*Modularity* adalah suatu ukuran dalam menghitung perbedaan antara *edge* dalam suatu *communities* yang diekstrak dari *network*. *Modularity* digunakan pada metode *Louvain* untuk mendeteksi *communities item* dari *network of product* (Verma, 2017).

Jika  $A_{vw}$  adalah suatu elemen dari matriks *adjacency* suatu *network*, maka:

$$A_{vw} = \begin{cases} 1 & \text{jika dan terhubung,} \\ 0 & \text{lainnya} \end{cases} \quad (2.9)$$

dan misal *nodes* terbagi ke dalam *communities*, sebagai contoh *node v* termasuk dalam *community cv*. Maka pecahan *edges* yang jatuh di dalam *communities*, sebagai contoh bahwa *nodes* yang terhubung dan berada di *community* yang sama adalah:

$$\frac{\sum_{vw} A_{vw} \delta(c_v, c_w)}{\sum_{vw} A_{vw}} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, c_w)$$

$$\delta(c_v, c_w) = \begin{cases} 1 & \text{jika } c_v = c_w, \\ 0 & \text{lainnya} \end{cases}$$

$$m = \frac{1}{2} \sum_{vw} A_{vw} \quad (2.10)$$

Dengan

$m$  : jumlah *edges* pada *graph*

Derajat  $k_v$  dari *node v* didefinisikan sebagai jumlah *edges* untuk kejadian di atasnya:

$$k_v = \sum_{vw} A_{vw} \quad (2.11)$$

Probabilitas suatu *edge* ada di antara *nodes v* dan *w* jika hubungan dibuat secara acak dengan memperhatikan derajat *node* adalah  $k_v k_w / 2m$ . Sehingga *modularity (Q)* didefinisikan sebagai:

$$Q = \frac{1}{2m} \sum_{vw} \left[ A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \quad (2.12)$$

*Modularity (Q)* dapat disederhanakan dengan persamaan-persamaan berikut:

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, i) \delta(c_w, j) \quad (2.13)$$

Dengan

$e_{ij}$  : pecahan *edges* yang *nodes*-nya tergabung dalam *community i* dan terhubung ke *community j*.

$$a_i = \frac{1}{2m} \sum_v k_v \delta(c_v, i) \quad (2.14)$$

Dengan

$a_i$  : pecahan ujung *edge* yang melekat pada *nodes* di *community i*

Maka,  $\delta(c_v, c_w) = \sum_i \delta(c_v, i) \delta(c_w, i)$  sehingga *modularity (Q)*

menjadi (Clauset, Newman, & Moore, 2004):

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{vw} \left[ A_{vw} - \frac{k_v k_w}{2m} \right] \sum_i \delta(c_v, i) \delta(c_w, i) \\ &= \sum_i \left[ \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, i) \delta(c_w, i) - \frac{1}{2m} \sum_v k_v \delta(c_v, i) \sum_w k_w \delta(c_w, i) \right] \\ &= \sum_i (e_{ij} - a_i^2) \end{aligned} \quad (2.15)$$

dengan

$e_{ij}$  : pecahan sejumlah *edge* yang menggabungkan sejumlah *node* pada *community ke-i*

$a_i$  : pecahan ekspektasi ujung *edge* pada *community ke-i*

### 2.3.4 Metode Louvain

Metode *Louvain* menemukan partisi dengan *modularity* yang tinggi dari *network* yang besar dalam waktu yang singkat dan membuka struktur *network* suatu *community* hierarki dengan memberikan akses kepada resolusi yang berbeda dari deteksi *community* (Verma, 2017). Algoritma metode ini terbagi dalam 2 fase yang diulang secara iteratif sebagai berikut ini.

1. Fase pertama adalah mengalokasikan *community* yang berbeda pada tiap *node* dari *network*. Pada partisi awal ini, banyaknya *community* sebanyak *node* yang ada. Kemudian, hitung *modularity* tiap *node i* terhadap tiap *node j*. Jika hasil yang didapatkan maksimum dan positif, maka *node i* dipindahkan dari *community* asal ke *community j*. Jika hasil yang didapatkan bukan positif, maka *i* tetap berada di *community* asal. Fase ini berhenti ketika *modularity* maksimum lokal tercapai.
2. Fase kedua adalah membangun *network* baru yang memiliki *nodes* yang sudah merupakan *community*. Untuk melakukannya, frekuensi antara *nodes* yang baru diberikan oleh jumlah berat dari *edge* antara titik-titik pada 2 *community* yang berkorespondensi. *Edge* antara titik-titik pada *community* yang sama membentuk lingkaran diri untuk *community* tersebut dalam *network* yang baru. Ketika fase kedua selesai, kemungkinan dapat terjadi pengulangan pada fase pertama atau iterasi.

### 2.3.5 Centrality

Pada *network analysis*, *centrality* mengidentifikasi *nodes* yang paling penting dalam suatu grafik, atau dalam kasus ini mendapatkan barang atau *item* yang paling penting di dalam *network*. Jika terdapat perpindahan dari satu *node* ke *node* lainnya melalui jalan terpendek, maka *node* yang banyak disinggahi memiliki *centrality* yang lebih tinggi (Verma, 2017). Salah satu metode untuk menentukan *centrality* adalah *degree centrality*. *Degree centrality* mengasumsikan bahwa *node* yang penting memiliki banyak koneksi (hubungan). Secara matematis, *degree centrality* suatu *node* pada suatu *graph* atau *network* adalah sebagai berikut (Lakha & Dhandhanian, 2018):

$$C_{\text{degree}} = \frac{d}{(N - 1)} \quad (2.16)$$

dengan

$N$  : jumlah *node* pada *network*

$d$  : jumlah koneksi (hubungan)

### 2.3.6 Page Rank

*Page Rank* diimplementasikan dengan menghitung jumlah hubungan (*link*) ke suatu *node* untuk menentukan seberapa pentingnya *node* tersebut. Semakin banyak *edge* yang diterima oleh suatu *node*, maka semakin penting *node* tersebut. Aspek-aspek untuk menentukan *PageRank* suatu *node* adalah jumlah *edge* yang diterima, *centrality* suatu *node* dari arah datangnya *edge*, dan kecenderungan hubungan (*link*) dari penghubung (*linker*) (Verma, 2017). *Page Rank* didefinisikan dengan persamaan berikut:

$$P(i) = \sum_{(j,i) \in E} \frac{P(j)}{O_j} \quad (2.17)$$

dengan

$P(j)$  : jumlah *node* yang ada pada *community* atau *graph*

$O_j$  : jumlah *edge* yang keluar dari *node* yang terhubung dengan *node j*

$E$  : kumpulan *edge* pada suatu *community* atau *graph*

