

BAB II

LANDASAN TEORI

2.1 PT. Telkomsel *Branch* Tasikmalaya

PT. Telkomsel *Branch* Tasikmalaya merupakan salah satu anak perusahaan yang bergerak dalam bidang telekomunikasi seluler. Perusahaan ini mempunyai beragam produk yang sesuai dengan kebutuhan segmentasi konsumen serta mengakuisisi tiga *cluster* (wilayah penjualan) di Priangan Timur yaitu Ciamis, Garut dan Tasikmalaya, serta berkontribusi atau bekerja sama dengan beberapa *Vendor, Mitra* dan *counter cellular* dalam pendistribusian pemasaran produk.

Divisi PT. Telkomsel *Branch* Tasikmalaya yang bergerak di lapangan untuk melakukan kunjungan *survei* serta mempromosikan *product knowledge* Telkomsel adalah Divisi *Broadband and Digital Sales* yang memiliki tugas utama untuk melakukan pendataan *outlet* dalam memantau pencapaian penjualan serta pembelian berupa produk kartu perdana kuota internet, kartu perdana segel, Omset M-Kios serta penjualan *device handphone* dan lainnya untuk mengetahui *market share* tingkat pangsa pasar penjualan berbagai produk operator lain yang telah didistribusikan (Pimpinan Divisi *Broadband and Digital Sales*. 2018).

2.2 Analisis Asosiasi (*Association Rule Mining*)

Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan asosiasi antara suatu kombinasi item. Analisis asosiasi dikenal juga sebagai salah satu teknik data mining yang menjadi dasar dari salah satu teknik data mining lainnya. (Kusrini & Luthfi, E.T., 2009). Bila kita

mengambil contoh aturan asosiatif dalam suatu transaksi pembelian barang disuatu minimarket adalah kita dapat mengetahui berapa besar kemungkinan seorang konsumen membeli suatu item bersamaan dengan item lainnya (membeli roti bersama dengan selai). Karena awalnya berasal dari studi tentang database transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama apa, maka *association rule* sering juga dinamakan *market basket analysis* (Ristianingrum & Sulastri, 2017).

Market basket analysis yaitu analisis yang sering dipakai untuk menganalisa isi keranjang belanja konsumen dalam suatu pasar swalayan. Contoh penerapan dari aturan asosiatif adalah analisa pembelian produk pada sebuah toko alat tulis, pada analisa itu misalkan dapat diketahui berapa besar kemungkinan seorang pelanggan membeli pensil bersamaan dengan membeli penghapus. Penerapan aturan asosiasi dalam kasus tersebut dapat membantu pemilik toko untuk mengatur penempatan barang, mengatur persediaan atau membuat promosi pemasaran dengan menerapkan diskon untuk kombinasi barang tertentu (Gama, Adie Wahyudi O. et al, 2016).

Bentuk umum dari *association rule* adalah *Antecedent* dan *Consequent*. Bila kita ambil contoh dalam sebuah transaksi pembelian barang di sebuah minimarket didapat bentuk *association rule* roti selai. Yang artinya bahwa pelanggan yang membeli roti ada kemungkinan pelanggan tersebut juga akan membeli selai, dimana tidak ada batasan dalam jumlah *item-item* pada bagian *antecedent* ataupun *consequent* dalam sebuah *rule* (Ristianingrum & Sulastri,2017).

Ide dari aturan asosiasi adalah untuk memeriksa semua kemungkinan hubungan *if-then* antar item dan memilih hanya yang paling mungkin (*most likely*) sebagai indikator dari hubungan ketergantungan antar *item*. Biasanya digunakan istilah *antedecent* untuk mewakili bagian “jika” dan *consequent* untuk mewakili bagian “maka”. Dalam analisis ini, *Antedecent* dan *consequent* adalah sekelompok item yang tidak punya hubungan secara bersama (Santoso, 2007). *Association rule* memiliki dua tahap pengerjaan, yaitu :

1. Mencari kombinasi yang paling sering terjadi dari suatu itemset.
2. Mendefinisikan *Condition* dan *Result* (untuk *conditional association rule*).

Dalam menentukan suatu *association rule*, terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapat dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu :

1. **Support** adalah 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 (misal, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa item A dan B dibeli bersamaan).
2. **Confidence** adalah suatu ukuran yang menunjukkan hubungan antar dua item secara *conditional* (berdasarkan suatu kondisi tertentu). Misalnya seberapa sering item B dibeli jika orang membeli item A.

Kedua ukuran ini nantinya berguna untuk menentukan kekuatan suatu pola dengan membandingkan pola dengan nilai minimum kedua parameter yang ditentukan oleh pengguna. Bila suatu pola memenuhi kedua nilai minimum

parameter yang sudah ditentukan sebelumnya, maka pola tersebut dapat disebut sebagai *interesting rule* atau *strong rule* (Ristianingrum & Sulastri, 2017).

Analisis asosiasi didefinisikan sebagai suatu proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk *support* dan syarat minimum untuk *confidence* (Gama, Adie Wahyudi O. et al, 2016).

2.3 Metodologi Dasar Aturan Asosiasi

Metodologi dasar aturan asosiasi terbagi menjadi dua tahap (Kusrini & Luthfi, E.T.,2009) yaitu :

1. Analisis Pola Frekuensi Tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* (penunjang) yaitu persentase item atau kombinasi item yang ada pada *database*. Nilai *support* sebuah item diperoleh dengan rumus berikut :

$$\mathbf{Support (A) = \frac{Jumlah\ Transaksi\ Untuk\ A}{Total\ Transaksi}} \quad (2.1)$$

Rumus support tersebut menjelaskan bahwa nilai *support* didapat dengan cara membagi jumlah transaksi yang mengandung item A (satu item) dengan jumlah total seluruh transaksi.

Sedangkan nilai *support* dari 2 item diperoleh dari rumus berikut:

$$\mathbf{Support (A, B) = P (A \cap B)} \quad (2.2)$$

$$\mathbf{Support (A, B) = \frac{\Sigma\ Transaksi\ Untuk\ A\ dan\ B}{\Sigma\ Transaksi}} \quad (2.3)$$

Rumus *support* diatas menjelaskan bahwa nilai *support* 2-itemsets didapat dengan cara membagi jumlah transaksi yang mengandung *item* A dan *item*

B (item pertama bersamaan dengan item yang lain) dengan jumlah total seluruh transaksi

2. Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung nilai *confidence* aturan asosiatif $A \rightarrow B$. Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dari rumus sebagai berikut:

$$\mathbf{Confidence} = P(B|A) = \frac{\Sigma \text{Transaksi Untuk A dan B}}{\Sigma \text{Transaksi A}} \quad (2.4)$$

Rumus diatas menjelaskan bahwa nilai *confidence* diperoleh dengan cara membagi jumlah transaksi yang mengandung *item* A dan *item* B (*item* pertama bersamaan dengan item yang lain) dengan jumlah transaksi yang mengandung *item* A (*item* pertama atau *item* yang ada disebelah kiri). Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan $Support \times Confidence$. Aturan diambil sebanyak n aturan yang memiliki hasil terbesar (Gama, Adie Wahyudi O. et al, 2016).

2.4 Algoritma Apriori

Algoritma *apriori* adalah salah satu algoritma dasar yang diusulkan oleh Agrawal dan Srikan pada tahun 1994 untuk menemukan *frequent itemsets* pada aturan asosiasi *Boolean*. Ide utama pada algoritma apriori adalah pertama, mencari *frequent itemset* (himpunan item-item yang memenuhi minimum *support*.) dari basis data transaksi, kedua menghilangkan *itemset* dengan frekuensi yang rendah berdasarkan level minimum *support* yang telah ditentukan sebelumnya. Selanjutnya membangun aturan asosiasi dari *itemset* yang memenuhi

nilai minimum *confidence* dalam basis data. Algoritma *apriori* adalah salah satu algoritma untuk melakukan pencarian *frequent itemset* dengan *association rules*. Algoritma *apriori* menggunakan pendekatan *level – wise search*, dimana *k-itemset* digunakan untuk memperoleh $(k+1)$ *itemset* (Aprianti, Winda et al 2017).

Sesuai dengan aturan asosiasi, algoritma *apriori* juga menggunakan minimum *support* dan minimum *confidence* untuk menentukan aturan *itemset* mana yang sesuai untuk digunakan dalam pengambilan keputusan (Gama et al, 2016). Menurut Han dan Kamber (2006), untuk membentuk kandidat *itemset* ada dua proses utama yang dilakukan algoritma *apriori* :

1. **Join Step** (Penggabungan) Pada proses ini setiap *item* dikombinasikan dengan *item* lainnya sampai tidak terbentuk kombinasi lagi.
2. **Prune Step** (Pemangkasan) Pada proses ini, hasil dari *item* yang dikombinasikan tadi kemudian dipangkas dengan menggunakan minimum *support* yang telah ditentukan oleh *user*.

Beberapa Istilah yang digunakan dalam algoritma *apriori* antara lain :

1. **Support** (dukungan) adalah probabilitas pelanggan membeli beberapa produk secara bersamaan dari seluruh transaksi. *Support* untuk aturan “ $X \Rightarrow Y$ ” adalah probabilitas atribut atau kumpulan atribut X dan Y yang terjadi bersamaan. Rumus mencari nilai *Support* :

$$\mathbf{Support (A)} = \frac{\mathbf{Jumlah\ Transaksi\ Mengandung\ A\ dan\ B}}{\mathbf{Total\ Transaksi}} \times \mathbf{100\%} \quad (2.5)$$

2. **Confidence** (tingkat kepercayaan): probabilitas kejadian beberapa produk dibeli bersamaan dimana salah satu produk sudah pasti dibeli. Contoh: Misalnya $A \rightarrow B$ memiliki nilai *confidence* 80%. Artinya jika membeli *item*

A, maka pasti membeli *item* B dimana nilai kepastiannya adalah 80%.

Rumus mencari *confidence* :

$$\text{Confidence (A)} = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi A}} \times 100\% \quad (2.6)$$

3. **Minimum support** adalah parameter yang digunakan sebagai batasan frekuensi kejadian atau *support count* yang harus dipenuhi suatu kelompok data untuk dapat dijadikan aturan.
4. **Minimum confidence** adalah parameter yang mendefinisikan minimum level dari *confidence* yang harus dipenuhi oleh aturan yang berkualitas.
5. **Itemset** adalah sekumpulan item yang memiliki kategori sama. Dikatakan harus memiliki kategori sama karena *item-item* ini yang akan dikombinasikan, dicari jumlah kemunculannya dalam data, dan ditarik kesimpulan dari kombinasi akhir yang memiliki nilai *asosiasi rule* terbesar.
6. **Frequent item set** adalah menunjukkan jumlah kemunculan sekumpulan item (yang telah ditentukan) dalam data
7. **Support count** adalah frekuensi kejadian untuk sebuah kelompok produk atau *itemset* dari seluruh transaksi.
8. **Kandidat itemset** adalah *itemset-itemset* yang akan dihitung *support count*-nya. Untuk membentuk kandidat *2-itemset* digunakan rumus kombinasi sebagai berikut :

$$C^n = \frac{n!}{n-3 \cdot 3!} \quad (2.7)$$

9. **Kombinasi 2 itemset (F2)** adalah Himpunan dari *frequent k-itemset* dilambangkan dengan F_k . F2 adalah bentuk kombinasi dari 2 *item*. Misal F2

adalah kombinasi {A, B} bernilai 5. Artinya kombinasi A dan B dalam data berjumlah 5.

10. **K-itemset** adalah *Itemset* yang terdiri dari kandidat item yang ada pada himpunan. Intinya Kandidat itu adalah jumlah unsur yang terdapat pada suatu himpunan.

2.5 Tahapan Perhitungan Algoritma Apriori

Algoritma apriori dibagi menjadi beberapa tahapan yang disebut *iterasi*. Pada *iterasi* pertama dihasilkan pola frekuensi tinggi dengan panjang *1-itemset*. Pada *iterasi* pertama pula, nilai *support* dari setiap *item* dihitung dengan melihat *database* yang ada. Setelah didapat nilai *support*, *item* yang memiliki *support* di atas minimum *support* (minsup) dipilih sebagai acuan untuk *iterasi* selanjutnya. *Iterasi* kedua menghasilkan *2-itemset* yang tiap setnya memiliki dua *item*. Tiap kandidat *2-itemset* dihitung nilai *support*-nya dengan melihat *database* sebelumnya. *Support* disini dimaksudkan untuk jumlah transaksi dalam *database* yang mengandung kedua item dalam kandidat *2-itemset*. Sama seperti *1-itemset*, setelah didapat nilai *support*-nya, maka kandidat *2-itemset* yang memenuhi minsup dapat ditetapkan sebagai *2-itemset* yang merupakan pola frekuensi tinggi (Destiyati, Orisky S.A. & Aribowo, Eko, 2015). Untuk selanjutnya pada *iterasi* ke *k-itemset* dapat dibagi menjadi beberapa bagian diantaranya :

1. Membentuk kandidat *itemset*, kandidat *k-itemset* dibentuk dari kombinasi *itemset* yang didapat dari *iterasi* sebelumnya (*join*). Satu ciri dari algoritma Apriori adalah adanya pemangkasan (*prune*) kandidat *k-itemset* yang *subset*-nya berisi *k-1 item* tidak termasuk dalam pola frekuensi tinggi

dengan panjang $k-1$. Penghitungan *support* dari tiap kandidat *k-itemset* *Support* dari tiap kandidat *k-itemset* didapat dengan melakukan *scan database* untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat *k-itemset* tersebut.

2. Menghitung *support* dari tiap kandidat *k-itemset*, *Support* dari tiap kandidat *k-itemset* didapat dengan men-*scan database* untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat *k-itemset* tersebut. Ini adalah juga ciri dari algoritma *apriori* dimana diperlukan penghitungan dengan *scan* seluruh *database* sebanyak *k-itemset* terpanjang.
3. Pola frekuensi tinggi yang memuat *k-item* atau *k-itemset* ditetapkan dari kandidat *k-itemset* yang *supportnya* lebih besar dari minimum *support*.
4. Bila tidak didapat pola frekuensi tinggi baru, maka seluruh proses dihentikan. Bila tidak, maka kandidat ditambah satu dan kembali ke bagian satu.