



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Data

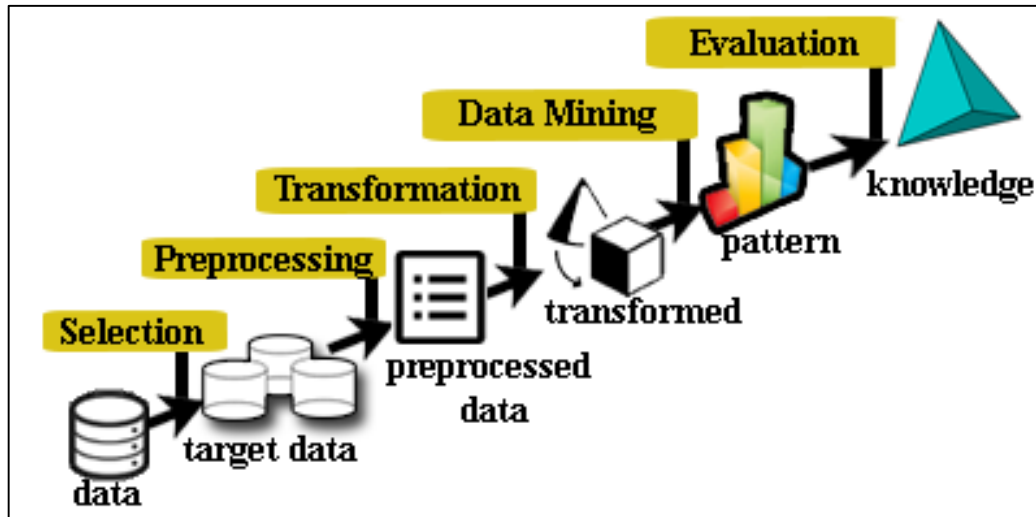
Data merupakan kata serapan dari Bahasa Inggris yang memiliki makna jamak, sedangkan datum bermakna tunggal. Arti kata data adalah segala sesuatu yang dapat diterima oleh indera manusia dari rangsangan yang ada di sekitar, baik tersurat maupun tersirat (Wahyudi, 2003).

Whitten dan Bentley (2007) menyatakan, data adalah sebuah sumber daya yang harus dikontrol dan dikelola menjadi suatu bentuk yang lebih berguna dan bermanfaat.

#### 2.2 Data Mining

*Data mining* adalah cara menyelesaikan sebuah masalah dengan menganalisis data yang ada dalam *database* (Whitten dkk, 2005). *Data mining* digunakan untuk menguraikan penemuan pengetahuan di dalam *database* (Turban dkk, 2007). Menurut Kusriani dan Luthfi (2009), *data mining* adalah proses yang menggunakan teknik Statistik, Matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari *database* yang besar.

Proses *Knowledge Discovery in Databases* (KDD) menurut Fayyad (1996), dalam jurnalnya yang berjudul “*From Data Mining to Knowledge Discovery in Databases*”, dapat diilustrasikan dengan gambar berikut (Fayyad dkk, 1996).



Gambar 2.1 Tahapan *Knowledge Discovery in Databases*  
(Sumber: Fayyad dkk, 1996)

Tahapan penyusunan KDD seperti yang digambarkan pada Gambar 2.1 umumnya didefinisikan sebagai berikut (Chandrawati, 2010).

1. *Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional. *Selection* merupakan proses pemilihan dan penyaringan *data* berdasarkan beberapa kriteria sesuai kebutuhan peneliti, misalkan berdasarkan jenis barang.

2. *Pre-processing dan Cleaning Data*

Mempersiapkan data dengan cara membersihkan data yang akan diproses serta menghapus informasi atau *field-field* yang tidak dibutuhkan, yang nantinya akan memperlambat kinerja proses *query*. Proses *cleaning* mencakup membuang duplikasi data, memeriksa data yang tidak konsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak.

### 3. *Transformation*

Proses ini mentransformasikan atau menggabungkan data ke bentuk yang lebih tepat untuk melakukan proses *mining* dengan cara melakukan peringkasan (agregasi).

### 4. *Data mining*

Proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan. Setidaknya ada tiga teknik yang paling sering digunakan pada *data mining*, yaitu: *classification*, *clustering* dan *association rule mining*.

### 5. *Interpretation and evaluation*

Proses untuk menerjemahkan pola-pola yang dihasilkan dari *data mining*. Mengevaluasi (menguji) apakah pola atau informasi yang ditemukan bersesuaian atau bertentangan dengan fakta atau hipotesa sebelumnya. Pengetahuan yang diperoleh dari pola-pola yang terbentuk dipresentasikan dalam bentuk visualisasi.

Dalam buku "*Discovering Knowledge in Data*" (Larose, 2005), *data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dilakukan, yaitu:

#### a) Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan *attribute* yang muncul dalam waktu yang sama.

#### b) Deskripsi

Deskripsi merupakan salah satu fungsi *data mining* untuk menemukan atau menggali sekumpulan data dalam jumlah yang banyak dengan menggambarkan

pola yang terjadi. Deskripsi memiliki kecenderungan memberikan penjelasan dari suatu pola.

c) Klasifikasi

Klasifikasi dalam *data mining* bekerja pada data historis atau data sejarah. Data historis bisa disebut sebagai data latih atau data *training*. Histori data digunakan sebagai cara untuk mendapatkan pengetahuan dan disebut data pengalaman. Dalam klasifikasi lebih fokus kepada target variabel kategori. Sebagai contoh, penggolongan penjualan dapat dikelompokkan dalam tiga kategori, yaitu penjualan tinggi, penjualan sedang, dan penjualan rendah.

d) Estimasi

Estimasi memiliki banyak persamaan dengan klasifikasi, kecuali variabel target estimasi lebih fokus kepada numerik dibandingkan ke arah kategori. Estimasi dapat digunakan untuk menerka sebuah nilai yang belum diketahui.

e) Pengklasteran

Pengklasteran merupakan pengelompokan dengan mengidentifikasi data dengan karakteristik tertentu. Berbeda dengan klasifikasi, pada pengklasteran tidak ada variabel target. Pengklasteran tidak melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan.

f) Prediksi

Prediksi memiliki kemiripan dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan

teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan untuk prediksi pada keadaan yang tepat.

### 2.3 Market Basket Analysis

Budhi dan Santosa (2007) menyatakan, *Market Basket Analysis* merupakan sebuah analisis terhadap perilaku konsumen berbelanja, dengan cara menemukan asosiasi di antara berbagai macam barang yang dimasukkan ke dalam keranjang belanja. *Market basket analysis* cocok diterapkan pada minimarket atau supermarket.

Informasi atau pengetahuan seperti di atas tentunya tidak hanya bermanfaat di dalam lingkungan pemasaran untuk pasaran swalayan saja. Beberapa bisnis yang bergerak di luar wilayah ini pun bisa menikmati manfaat dari adanya *Market Basket Analysis* ini. Sebagai contoh, toko virtual yang menjual produknya secara *online*, bank yang memberikan fasilitas layanan kartu kredit untuk para nasabahnya, perusahaan penyedia jasa asuransi, restoran *fastfood*, toko baju, toko buku, dan lain-lain.

Tujuan dari penggunaan metode *market basket analysis* adalah untuk mengetahui produk apa saja yang dibeli oleh konsumen dalam waktu yang bersamaan. Dengan menganalisis data transaksi dapat ditemukan pola pembelian barang. Informasi tersebut kemudian dapat digunakan untuk membantu manager atau petinggi perusahaan untuk meningkatkan penjualan melalui promosi produk yang bervariasi.

Beberapa kasus dari pola penjualan barang dalam satu transaksi oleh konsumen mudah untuk ditebak, misalnya susu dibeli bersamaan dengan roti. Namun, mungkin saja terdapat suatu pola pembelian barang-barang yang tidak

pernah terpikirkan sebelumnya. Misalnya, pembelian minyak goreng dengan deterjen. Mungkin saja pola seperti ini tidak pernah terpikirkan sebelumnya karena minyak goreng dan deterjen tidak mempunyai hubungan sama sekali, baik sebagai barang pelengkap maupun barang pengganti. Hal ini mungkin tidak pernah terpikirkan sebelumnya sehingga tidak dapat diantisipasi jika terjadi sesuatu, seperti kekurangan stok deterjen misalnya. Contoh tersebut merupakan salah satu manfaat yang dapat diperoleh dengan melakukan *Market Basket Analysis*. Jika proses ini dilakukan secara terkomputerisasi, maka seorang manajer tidak perlu mengalami kesulitan untuk menemukan pola barang apa saja yang mungkin dibeli secara bersamaan (Gunawan, dkk, 2015).

#### 2.4 Algoritma Apriori

Algoritma Apriori adalah algoritma dasar yang diusulkan oleh Agrawal dan Srikant pada tahun 1994 untuk menentukan nilai *frequent itemset* dalam aturan asosiasi *boolean* (Han, Kamber, dan Pei, 2002). Beberapa parameter untuk mencari aturan asosiasi menggunakan algoritma Apriori adalah *minimum support* dan *minimum confidence*. Kedua parameter tersebut menjadi acuan untuk menemukan *frequent* baru yang memiliki *strong rule*. *Minimum support* merupakan ukuran yang menunjukkan besar tingkat dominasi atau *presentase* suatu kombinasi *item* dari keseluruhan transaksi (Kusrini dan Luthfi, 2009). *Minimum confidence* merupakan ukuran atau nilai kepastian yang menunjukkan kuatnya hubungan antar *item* atau barang. Sebagai contoh, seberapa sering *item* A dibeli jika pelanggan lain membeli *item* B. Pola yang terbentuk disebut *interesting rule* atau *strong rule*. *Strong rule* ditemukan bila pola memenuhi kedua nilai minimum yang telah ditetapkan.

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

a. Analisa Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support item* diperoleh dengan persamaan sebagai berikut (Larose, 2005).

$$Support(A) = \left( \frac{\sum \text{Transaksi yang mengandung } A}{\sum \text{Transaksi}} \right) \dots \text{Rumus (2.1)}$$

Nilai *support* dari 2 *item* diperoleh dari persamaan berikut.

$$Support(A, B) = \left( \frac{\sum \text{Transaksi yang mengandung } A \text{ dan } B}{\sum \text{Transaksi}} \right) \dots \text{Rumus (2.2)}$$

b. Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tinggi ditemukan, kemudian mencari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif  $A \rightarrow B$ . Nilai tersebut diperoleh dari persamaan berikut (Larose, 2005).

$$Confidence = P(B|A) = \left( \frac{Support(A, B)}{Support(A)} \right) \dots \text{Rumus (2.3)}$$

Untuk menemukan *frequent itemset* dengan menggunakan algoritma Apriori adalah sebagai berikut (Erwin, 2009).

1. Penggabungan

Proses ini membuat kombinasi semua *item* sampai tidak terbentuk kombinasi lagi.

2. Pemangkasan

Pada proses ini *item* yang telah dikombinasikan, dipangkas dengan melakukan pemeriksaan dengan minimum support yang telah ditentukan.



Untuk menjelaskan langkah-langkah dalam menemukan *itemset* berikut adalah tahapan yang harus dilaksanakan (Setiawati, 2009).

1. Pembentukan kandidat *itemset*, *k-itemset* dibentuk dari kombinasi  $(k-1)$ -*itemset* yang didapat dari iterasi sebelumnya.
2. Penghitungan *support* dari setiap kandidat *k-itemset*. *Support* dari tiap kandidat *k-itemset* didapat dengan melakukan *query database* untuk menghitung jumlah transaksi yang memuat semua *item* dalam kandidat *k-itemset* tersebut.
3. Tetapkan pola frekuensi tinggi. Pola frekuensi tinggi yang memuat *k-item* atau *k-itemset* ditetapkan dari kandidat *k-itemset* yang nilai *support* lebih dari *minimum support*.
4. Jika tidak didapatkan pola frekuensi tinggi baru maka seluruh proses dihentikan.

## 2.5 Algoritma Apriori Shankar Bargadiya

Shankar dan Bargadiya melakukan perbaikan terhadap algoritma apriori yang sudah ada sebelumnya. Algoritma ini mengatasi kelemahan utama pada Apriori, yaitu *query database* yang berulang-ulang sehingga berpengaruh pada kecepatan proses data. Algoritma Improved Apriori Shankar Bargadiya hanya memerlukan *scan database* sebanyak dua kali (Chenly, 2014).

Aktivitas yang dilakukan pertama kali adalah menemukan *frequent 1-itemset* dari *database* kemudian menginisialisasi variabel *count* = 0. Inisialisasi tersebut dimasukkan ke dalam *Global power set*. Ketika *query database* dilakukan bertujuan untuk menghitung *itemset*, *itemset* yang tidak memenuhi *minimum support* akan dihapus. Langkah ini mengurangi generasi berlebihan dari kandidat *item* yang akan

dihasilkan. Setelah itu, mencari *local power set*, yaitu hasil penghitungan *item* yang tersisa dari *Global power set*. Langkah ini mengurangi *query database* yang berulang (Shankar dan Bargadiya, 2013b).

Berikut adalah langkah kerja algoritma Apriori yang dikembangkan oleh Shankar dan Bargadiya (2013b).

1. L1 = digunakan untuk menemukan *frequent 1-itemset*.
2. Membuat *global power set*, yaitu L1 dengan inisialisasi *count* = 0.
3. Melakukan *query database* hingga transaksi terakhir.
  - a. Membaca *itemset* dari transaksi dan hapus *item* yang tidak ada di L1.
  - b. Memeriksa *global power set* dan *local power set*. Jika sama, maka *count global power set* akan ditambah.
4. Memeriksa *global power set* dan perhatikan setiap *item*. Jika di bawah *minimum support*, maka *item* tersebut dihapus dari *global power set*.
5. *Item* yang tersisa adalah *frequent itemset*.

Berikut adalah gambaran langkah kerja algoritma Apriori Shankar Bargadiya dengan *minimum support* = 3.

Tabel 2.1 Daftar Transaksi  
(Sumber: Chenly, 2014)

Tid (ID Transaksi )	Itemset
1	Item2, Item4
2	Item1, Item2, Item3, Item4
3	Item2, Item3
4	Item2, Item3
5	Item1, Item2, Item3
6	Item2
7	Item1, Item2
8	Item1, Item3
9	Item2, Item3
10	Item2

Langkah 1 : menemukan *frequent 1-itemset* dari *database*.

Tabel 2.2 *Frequent 1-itemset*  
(Sumber: Chenly, 2014)

Items	Frequency of Itemset
Item1	4
Item2	8
Item3	7

Item4 dihapus karena memiliki *count* = 2 sehingga tidak memenuhi *minimum support* yang telah ditentukan.

Langkah 2: Langkah ini membuat *Global Power Set* dengan inisialisasi variabel *count* = 0.

Tabel 2.3 Inisialisasi *Global Power Set*  
(Sumber: Chenly, 2014)

Candidate Item 1 itemset	Item1	Item2	Item3
1-Count	0	0	0
Candidate 2 itemset	Item1, Item2	Item1, Item3	Item2, Item3
2-Count	0	0	0
Candidate 3 itemset	Item1, Item2, Item3		
3-count	0		

Langkah 3: melakukan *query* data transaksi pada *database*.

Setelah *query database* selesai dilakukan, kemudian data tersebut akan dimasukkan ke dalam *Global Power Set*. *Global Power Set* menampung semua *data* yang telah diproses, kemudian terbentuklah *Global Power Set* sebagai berikut.

Tabel 2.4 Kandidat *itemset* dan frekuensi kemunculan *item*  
(Sumber: Chenly, 2014)

Candidate Item 1 itemset	Item1	Item2	Item3
1-Count	4	9	6
Candidate 2 itemset	Item1, Item2	Item1, Item3	Item2, Item3
2-Count	3	3	5
Candidate 3 itemset	Item1, Item2, Item3		
3-count	2		

Langkah 4: Berdasarkan Tabel 2.4, kemudian dilakukan pemeriksaan *minimum support* yang telah ditetapkan. Jika ada kandidat *itemset* yang tidak memenuhi syarat maka akan dilakukan penghapusan kepada *itemset* tersebut.

Tabel 2.5 Pemangkasan kandidat *itemset*  
(Sumber: Chenly, 2014)

Candidate Item 1 itemset	Item1	Item2	Item3
1-Count	4	9	6
Candidate 2 itemset	Item1, Item2	Item1, Item3	Item2, Item3
2-Count	3	3	5
Candidate 3 itemset	<b>Item1, Item2, Item3</b>		
3-count	<b>2</b>		

Langkah 5: Setelah pemangkasan selesai, terbentuk hasil *frequent itemset* yang ditampilkan pada tabel 2.6

Tabel 2.6 *Frequent Itemset*  
(Sumber: Chenly, 2014)

<b>Candidate Item 1 itemset</b>	<b>Item1</b>	<b>Item2</b>	<b>Item3</b>
1-Count	4	9	6
<b>Candidate 2 itemset</b>	<b>Item1, Item2</b>	<b>Item1, Item3</b>	<b>Item2, Item3</b>
2-Count	3	3	5

Candidate itemset yang tersisa, yaitu Item1 dan Item2, Item1 dan Item3, dan Item2 dan Item3 merupakan kombinasi *item* yang direkomendasikan.

## 2.6 Lift Ratio

*Lift Ratio* digunakan untuk mengevaluasi kuat tidaknya sebuah aturan asosiasi. *Lift Ratio* mengukur seberapa akurat pola atau *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*, nilai tersebut menunjukkan keabsahan informasi yang dihasilkan (Santosa, 2007).

Nilai *ratio* dapat dihitung menggunakan rumus sebagai berikut (Han dkk, 2004).

$$\text{Lift Ratio} = \left( \frac{\text{Support}(A, B)}{\text{Support}A * \text{Support}B} \right) \quad \dots \text{ Rumus (2.4)}$$

*Rule* yang *valid* adalah *rule* yang memiliki nilai *lift ratio* > 1 dengan menghitung nilai *support* yang mengandung A dan B dibagi dengan nilai *support* A \* nilai *support* B. *Rule* yang akan *dihitung* nilai *lift ratio*-nya merupakan *rule* yang memenuhi nilai *minimum support* yang ditetapkan.

