BAB III
LANDASAN TEORI

3.1 Data *Mining*

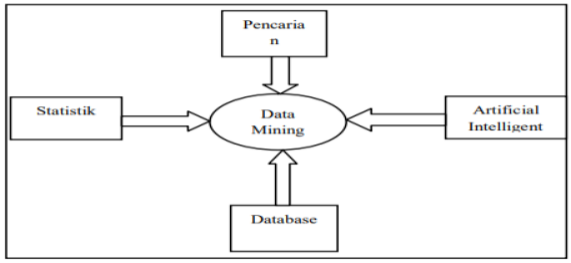
Data *mining* merupakan salah satu langkah pengetahuan dalam *database* yang bertujuan untuk mengumpulkan informasi yang bermanfaat. Data mining juga merupakan suatu proses penambangan informasi dari suatu data. Proses ini menggunakan *artificial intelligence* (AI), *machine learning*, serta teknik statistik. Dalam data *mining* terdapat berbagai teknik data *mining* utama yang telah dikembangkan serta diterapkan dalam proyek diantaranya klasifikasi, pengelompokan, dan aturan asosiasi [9]. Data *mining* juga sangat membantu perusahaan untuk mendapatkan pola dari data yang tersimpan di dalam basis data perusahaan. Keterlibatan manusia sangat dibutuhkan dalam setiap tahap proses data *mining* itu sendiri. Pemahaman terhadap model statistik dan matematik yang digunakan dalam perangkat lunak sangat dituntut [10].

Dari defenisi yang telah disampaikan, terdapat beberapa hal penting terkait dengan data *mining* yaitu [11]:

1. Data *mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang digunakan berupa data yang sangat besar.
3. Tujuan data *mining* yaitu mendapatkan hubungan atau pola sehingga memberikan indikasi yang bermanfaat.

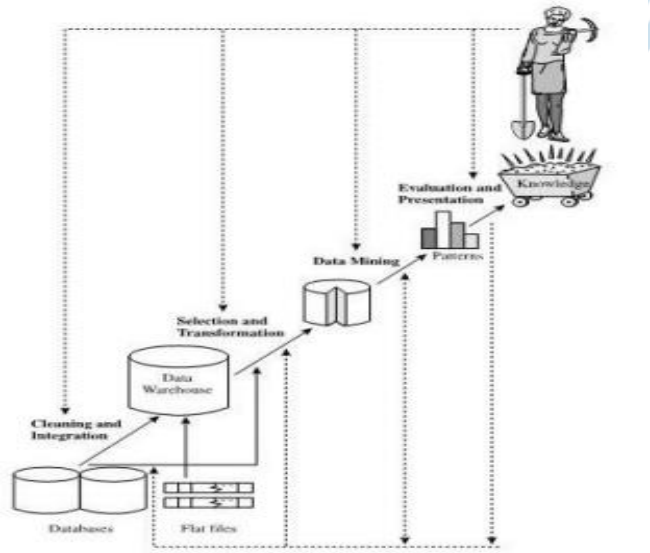
Data *mining* memiliki ilmu-ilmu yang berkaitan seperti *artificial intelligence*, *statistic*, *database* serta *machine learning*. Data *mining* merupakan bagian dari proses *Knowledge Discovery from Data* (KDD), seperti dapat dilihat pada gambar 3.1. *Knowledge discovery in database* (KDD) adalah kegiatan yang meliputi pengumpulan data, pembersihan data,

pemakaian data historis untuk menemukan keteraturan pola atau hubungan dalam set data berukuran besar [12].



Gambar 3. 1 Bidang Ilmu Data *Mining*

Proses KDD dalam data *mining* dimulai dari data mentah dan berakhir dengan informasi yang telah diolah. Proses tersebut dimulai dari proses *cleaning* sehingga menemukan data *warehouse.* Melakukan proses *selection* dan *transformation* yang kemudian disebut sebagai data *mining* hingga menemukan pola dan memperoleh pengetahuan dari data (*knowledge*) [13], seperti yang tampak dalam gambar 3.2.



Gambar 3. 2 Proses KDD dalam data *mining*

Adapun proses dari data *mining*, sebagai berikut:

1. Data *selection*

   Dalam proses ini melakukan data 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.

2. *Pre-processing/Cleaning*

   Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak. Dalam tahap ini perlu juga melakukan proses *encrihment*, dimana proses memperkaya data yang sudah ada dengan data lain yang relevan.

3. *Transformation*

   Proses *transformation* merupakan proses transformasi data yang sudah dipilih ke dalam bentuk mining. *Coding* merupakan proses *transformation* pada data yang terpilih. Proses *coding* dalam KDD merupakan proses kreatif dan sangat bergantung pada jenis pola informasi yang akan dicari dalam basis data.

4. Data *Mining*

   Data *mining* adalah proses yang paling penting dimana akan dilakukan berbagai teknik yang diaplikasikan untuk mancari pola atau informasi menarik dalam data yang terpilih. Metode atau algoritma dalam data *mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat tergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Knowledge Presentation*

   Proses tahap terakhir, dalam hal ini digunakan teknik visualisasi yang bertujuan untuk menginterpretasikan hasil dari data *mining*. Proses ini juga mencakup proses pemeriksaan apakah pola atau informasi yang

ditemukan bertentangan dangan fakta atau hipotesis yang ada sebelumnya [14].

Data *mining* dibagi atas beberapa kelompok berdasarkan tugas yang dapat dilakukan [11] :

1. Prediksi

   Dalam prediksi nilai dari hasil akan ada dimasa mendatang. Contoh prediksi adalah Prediksi harga cabe dalam 6 bulan yang akan datang.

2. Klasifikasi

   Klasifikasi memiliki target variabel kategori. Sebagai contoh penggolongan pendapatan pegawai dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

3. Pengklusteran

   Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record–record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Algoritma pengklusteran mencoba melakukan pembagian terhadap keseluruhan data menjadi sebuah kelompok yang memiliki kemiripan (*homogeny*), yang mana kemiripan dalam satu kelompok akan bernilai maksimal. Contoh pengklasteran adalah melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

4. Asosiasi

   Asosiasi bertugas menemukan pola yang muncul dalam banyak transaksi, dimana dalam transaksi tersebut terdiri dari beberapa item. Sebagai contoh melakukan analisis pembelian di

Pasar Swalayan untuk mengetahui seberapa besar kemungkinan seseorang pelanggan untuk membeli roti bersamaan dengan susu

3.2 Langkah-langkah Proses Aturan Asosiasi

Proses aturan Asosiasi terdiri dari beberapa tahap sebagai berikut:

1. Sistem men-*scandatabase* untuk mendapat kandidat 1-*itemset* (himpunan item yang terdiri dari 1 item) dan menghitung nilai *support*nya. Kemudian nilai *support*nya tersebut dibandingkan dengan minimum *support* yang telah ditentukan jika nilainya lebih besar atau sama dengan minimum *support* maka *item*set tersebut termasuk dalam *frequent*.
2. *Item*set yang tidak termasuk dalam *frequent* tidak diikutkan dalam iterasi selanjutnya.
3. Pada iterasi kedua sistem akan menggunakan hasil *frequent* pada iterasi pertama (F1) untuk membentuk kandidat pada iterasi kedua (F2). Pada iterasi selanjutnya sistem akan menggunakan hasil *frequent* pada iterasi selanjutnya akan menggunakan hasil *frequent* pada iterasi sebelumya (C1) untuk membentuk kandidat *item*set berikut (C). Sistem akan menggabungkan C1 dengan C1 untuk mendapatkan C2 seperti pada iterasi sebelumnya sistem akan menghapus kombinasi *item*set yang tidak termasuk dalam *frequent*.
4. Setelah dilakukan operasi *join*, maka pasangan *itemset* baru hasil proses *join* tersebut dihitung *support*nya.
5. Proses pembentuk kandidat yang terdiri dari proses *join* dan *pruning* akan terus dilakukan hingga himpunan kandidat itemnya *null*, atau sudah tidak ada lagi kandidat yang akan dibentuk.
6. Setelah itu, dari hasil *frequent itemset* tersebut dibentuk *association rule* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.
7. Pada pembentukan *association rule*, nilai yang sama dianggap sebagai suatu nilai.

8. *Association rule* yang terbentuk harus memenuhi nilai minimum yang telah ditentukan.
9. Untuk setiap *frequent* F, dicari himpuan bagian F yang tidak kosong. Untuk setiap himpunan bagian tersebut, dihasilkan *asssociation rules* yang memenuhi syarat *minimum* dengan menghitung *confidence association rules* A-> B [15].

3.3 Algoritma Apriori

Agrawal & Srikant merupakan orang yang memperkenalkan algoritma Apriori [16]. Konsep utamanya adalah untuk menghilangkan *itemset* dengan *support* kurang dari *minimum support*. *Support* dari *itemset* tidak pernah melebihi *support* dari himpunan bagiannya, properti ini dikenal sebagai properti *Anti-Monoton* [17]. Algoritma Apriori adalah salah satu algoritma yang dapat digunakan pada penerapan *market basket analysis* untuk mencari aturan-aturan asosiasi yang memenuhi batas *support* dan *confidence*. *Support* adalah presentase kombinasi item tersebut dalam *database*, sedangkan *confidence* adalah kuatnya hubungan antara-item dalam aturan asosiasi [11]. Algoritma apriori dibagi menjadi beberapa tahap sebagai berikut:

1. Pembetukan kandidat *itemset*.
2. Kandidat k-*itemset* dibentuk dari kombinasi (k-1)-*itemset* yang didapat dari iterasi sebelumnya. Satu cara dari algoritma apriori adalah pemangkasan kandidat k-*itemset* yang subsetnya berisi k-1 item tidak termasuk dalam pola frekuensi tinggi dengan panjang k-1.
3. Penghitungan *support* dari tiap kandidat k-*itemset*. *Support* dari tiap kandidat k-*itemset* didapat dengan men-*scan database* untuk mengitung jumlah transaksi yang memuat semua item didalam kandidat k-*itemset* tersebut. Ini adalah ciri dari algoritma apriori dimana diperlukan penghitungan dengan cara seluruh *database* sebanyak k-*itemset* terpanjang.

4. Tetapkan pola frekuensi tinggi. Pola frekuensi tinggi yang memuat k-item atau k-*itemset* ditetapkan dari kandidat dari k-*item*set yang *support*nya lebih besar dari minimum *support*.
5. Bila tidak didapat pola frekuensi tinggi baru maka seluruh proses dihentikan [15].

Pada pencarian aturan asosiasi membutuhkan parameter agar aturan yang didapat akurat. Parameter yang digunakan untuk membentuk suatu *rules* adalah:

a. *Support*

Support adalah suatu ukuran yang menunjukkan persentase kombinasi barang dari keseluruhan transaksi yang ada. Untuk menghitung *prosentase support* menggunakan rumus :

$$Support(A) = \frac{\sum \text{transaksi mengandung A}}{\sum \text{total transaksi}} \times 100\% \ldots\ldots\ldots [2]$$

b. *Confidence*

Confidence adalah suatu ukuran yang menunjukkan hubungan kondisional antara dua barang. Untuk menghitung nilai *confidence*, didapat dengan menggunakan rumus :

$$Confidence\ (A|B) = \frac{\sum \text{transaksi mengandung A } dan\ B}{\sum \text{transaksi mengandung A}} \times 100\% \ldots. [2]$$

Terdapat dua proses utama dalam algoritma apriori yaitu :

a. *Join* (penggabungan)

Dalam proses ini, setiap item dikombinasikan dengan *item* lain sampai tidak dapat terbentuk kombinasi lagi.

b. *Pruning* (pemangkasan)

Pada proses ini, hasil kombinasi *item* akan dipangkas berdasarkan *minimum support* yang telah ditentukan [6].

Contoh data uji sebuah toko dengan label id transaksi serta *items* dapat dilihat pada tabel 3.1.

Tabel 3. 1 Contoh Data Uji

Id Transaksi	*Items*
1	A,B
2	A,B,C,D
3	A,B,C,D,E
4	A,B,F
5	C,D,G

Pada tabel 3.2 menentukan *minimum support* yaitu 40%. Maka untuk iterasi 1 hitung dan pindai *database* agar mendapatkan sebuah pola dari *support*.

Tabel 3. 2 Contoh 1-*itemset*

Itemset	*Support Count*	*Support*
A	4	80%
B	4	80%
C	3	60%
D	3	60%
G	1	20%
E	1	20%
F	1	20%

Setelah itu hapus data yang tidak memenuhi *minimum support* hingga menemukan hasil yang ada pada tabel 3.3

Tabel 3. 3 Contoh Pola Frekuensi 1-*itemset*

Itemset	*Support Count*	*Support*
A	4	80%
B	4	80%
C	3	60%
D	3	60%

Kemudian ke iterasi kedua, iterasi sebelumnya pola frekuensi dari *support* telah didapat dari 1-*itemset*. Maka untuk mencari *k-itemset* menggunakan kombinasi yang ada pada tabel 3.4.

Tabel 3. 4 Contoh kombinasi 2-*itemset*

Itemset
A,B
A,C
A,D
B,C
B,D
C,D

Barulah melakukan iterasi kedua yang dinamakan C2 yang dihitung masing-masing frekuensi item dan didapatkan hasil seperti pada tabel 3.5.

Tabel 3. 5 Contoh 2-*itemset*

Itemset	*Support Count*	*Support*
A,B	4	80%
A,C	2	40%
A,D	2	40%
B,C	2	40%
B,D	2	40%
C,D	3	60%

Seletah itu memangkas k-*itemset* dengan menghitung *support* dari *itemset*, dilanjutkan dengan melakukan iterasi ketiga yang dapat kita lihat pada tabel 3.6.

Tabel 3. 6 Contoh Kombinasi 3-*itemset*

Itemset
A,B,C
A,B,D
B,C,D

Kandidat dari 3-*itemset* yang telah memenuhi *minimum support* maka akan menjadi panutan k-*itemset* yang hasilnya dapat dilihat pada tabel 3.7.

Tabel 3. 7 Contoh 3-*itemset*

Itemset	*SupportCount*	*Support*
A,B,C	2	40%

A,B,D	2	40%
B,C,D	2	40%

Setelah itu lakukan iterasi ke 4 karena jumlah *support* yang sama maka hasilnya ditunjukan pada tabel 3.8.

Tabel 3. 8 Contoh kombinasi 4-*itemset*

Itemset
A,B,C,D

Iterasi berhenti dikarenakan tidak adanya lagi kombinasi yang ditemukan, sehingga pola *frequent* tinggi yang ditemukan adalah "A,B,C,D". Pembentukan *asssociation rules* yang memenuhi syarat *minimum* dengan menghitung *confidence association rules* A-> B.

Tabel 3. 9 Aturan Asosiatif

Jika membeli (A)	Maka membeli (B)	*Support* (AUB)	*Support* (A)	*Confidence*
A,B,C	D	40%	40%	100%
A,B,D	C	40%	40%	100%
A,C,D	B	40%	40%	100%
B,C,D	A	40%	40%	100%
A,B	C	40%	80%	50%
A,C	B	40%	40%	100%

B,C	A	40%	40%	100%
A,B	D	40%	40%	100%
A,D	B	40%	40%	100%
B,D	A	40%	40%	100%
B,C	D	40%	40%	100%
B,D	C	40%	40%	100%
C,D	B	40%	60%	60,7%
A	B	80%	80%	100%
A	C	40%	80%	50%
A	D	40%	80%	50%
B	C	40%	80%	50%
B	D	40%	80%	50%
C	D	60%	60%	100%

Pembentukan aturan asosiatif sangatlah penting agar mendapat nilai *confidence*.