BAB II
DASAR TEORI

Bab ini menjelaskan tentang dasar teori yang digunakan dalam penyusunan tugas ahir ini, yaitu tentang pengertian *data mining,* pengelompokan *data mining,* teknik Asosiasi dan juga algoritma Apriori

2.1 *Data Mining*

Data Mining merupakan suatu proses untuk menemukan informasi atau pengetahuan yang berguna dari suatu data yang berjumlah besar. *Data mining* juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data (Pramudiono, 2007).

Saat ini istilah *data mining* sudah berkembang jauh dalam mengadaptasi setiap bentuk analisa data. Pada dasarnya *data mining* berhubungan dengan analisa data dan penggunaan teknik-teknik perangkat lunak untuk mencari pola dan keteraturan dalam himpunan data yang sifatnya tersembunyi. Teknik-teknik ini tentu bervariasi dan disesuaikan dengan kebutuhan analisis data.

2.1.1 Pengelompokan *Data Mining*

Terdapat tiga metode utama dalam *data mining* yaitu [9]:

a. *Supervised Learning* (Pembelajaran dengan Guru)

Algoritma ini melakukan proses belajar berdasarkan nilai dari variabel target yang terasosiasi dengan nilai dari variabel prediktor, biasanya sebagian besar algoritma *data mining* termasuk dalam metode ini antara lain *estimation, prediction/forecasting,* dan *classification.*

b. *Unsupervised Learning* (Pembelajaran tanpa Guru)

Algoritma ini mencari pola dari semua variable (atribut), algoritma *data mining* yang termasuk dalam metode ini adalah algoritma *clustering*.

c. *Association Learning* (Pembelajaran untuk Asosiasi Atribut)

Proses learning pada algoritma asosiasi (*association rule*) agak berbeda karena tujuannya adalah untuk mencari atribut yang muncul bersamaan dalam satu transaksi, biasanya metode ini banyak digunakan untuk analisa transaksi belanja. Algoritma *data mining* yang termasuk dalam metode ini adalah algoritma Apriori, *Fp. Growth.*

2.1.2 Tahap-tahap *Data Mining*

Terdapat 7 tahap utama *data mining* yaitu [6] :

1. Pembersihan Data (*Data Cleaning*)

Pembersihan data merupakan proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa *data mining* yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performasi dari teknik *data mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi Data (*Data Integration*)

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau file teks. Integrasi data dilakukan pada atribut-aribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

3. Seleksi Data (*Data Selection*)

Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*. Sebagai contoh, sebuah kasus yang meneliti faktor kecenderungan orang membeli dalam kasus *market basket analysis*, tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

4. Transformasi Data (*Data Transformation*)

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa metode standar seperti analisis asosiasi dan *clustering* hanya bisa menerima input data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi menjadi beberapa interval. Proses ini sering disebut transformasi data.

5. Proses *Mining*

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi Pola (*Pattern Evaluation*)

Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan. Dalam tahap ini hasil dari teknik *data mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai dengan hipotesa, maka ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki proses *data mining*, mencoba metode *data mining* lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

7. Presentasi Pengetahuan (*Knowledge Presentation*)

Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir dari proses *data mining* adalah bagaimana memformulasikan keputusan

atau aksi dari hasil analisis yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami *data mining*. Karenanya presentasi hasil *data mining* dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses *data mining*. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil *data mining* (Han, 2006).

2.2 Metode Asosiasi

Aturan asosiasi atau sering disebut *Association Rule*, merupakan salah satu metode yang digunakan untuk menemukan hubungan diantara data atau bagaimana suatu kelompok data mempengaruhi suatu keberadaan data lain. Aturan asosiasi merupakan salah satu metode yang sering digunakan untuk mencari hubungan antara berbagai *item*. Sebagai contohnya, dari sebuah himpunan data transaksi mungkin akan ditemukan kejadian berikut, dimana seseorang yang membeli susu dalam transaksi yang sama akan membeli gula, atau seseorang yang membeli sikat gigi dalam yang transaksi yang sama juga akan membeli odol.

Metode asosiasi dikenal sebagai salah satu metode *data mining* yang menjadi dasar dari berbagai metode *data mining* lainnya. Salah satu tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi (*frequent pattern mining*) menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu prosentase kombinasi item tersebut. dalam *database* dan *confidence* (nilai kepastian) yaitu kuatnya hubungan antar item dalam aturan assosiatif. Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan assosiatif yang memenuhi syarat minimum untuk support (*minimum support*) dan syarat minimum untuk *confidence* (*minimum confidence*) (Pramudiono, 2007).

Ada beberapa algoritma yang sudah dikembangkan mengenai aturan asosiasi, Metode ini terbagi atas beberapa algoritma yaitu algoritma Apriori, algoritma FP-Growth dan algoritma CT-Pro. Apriori merupakan algoritma klasik yang sering dipakai. Ide dasar dari algoritma ini adalah dengan mengembangkan *frequent itemset*, yaitu dengan cara menggunakan satu item dan kemudian secara rekursif mengembangkan *frequent itemset* tersebut dengan dua item, tiga item dan seterusnya hingga *frequent itemset* dengan semua ukuran. Untuk mengembangkan *frequent set* dengan dua item, dapat menggunakan *frequent set item*. Alasannya adalah bila set satu item tidak melebihi *support minimum,* maka sembarang ukuran itemset yang lebih besar tidak akan melebihi *support minimum* tersebut. Secara umum, mengembangkan set dengan fc-item menggunakan frequent set dengan k – 1 item yang dikembangkan dalam langkah sebelumnya. Setiap langkah memerlukan sekali pemeriksaan ke seluruh isi *database* [6].

2.3 Algoritma Apriori

Algoritma Apriori termasuk jenis aturan asosiasi pada data mining. Selain apriori, yang termasuk pada golongan ini adalah metode *generalized rule induction* dan algoritma *hash based*. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut *affinity analysis* atau *market basket analysis* [10].

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap yaitu[4]:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai support dalam *database*.

2. Pembentukan aturan assosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A ->B.

Ide dasar dari algoritma Apriori adalah mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam basis data. Nilai *support* sebuah *item* diperoleh dengan menggunakan rumus berikut[3]:

$$Support\ (A) = \frac{Jumla\ h\ Transaksi\ \ Mengandung\ \ A}{Total\ Transaksi} \quad \text{.....(i)}$$

Nilai *support* dari 2 *item* diperoleh dengan menggunakan rumus[3] :

$$Support\ (A,B) = P\ (A \cap B) \quad \text{.....(ii)}$$

$$Support\ (A,B) = \frac{\sum Transaksi\ \ Mengandung\ \ A\ dan\ B}{\sum Transaksi} \quad \text{.....(iii)}$$

Frequent itemset menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang ditentukan (Φ). Misalkan Φ = 2, maka semua *itemsets* yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent*. Himpunan dari *frequent* k-*itemset* dilambangkan dengan Fk[3].

Bentuk algoritma dari Apriori dapat dituliskan sebagai berikut[3]:

```
L1 = {frequent itemset with one element}
for (k=2; Lk-1≠∅; k++)
{
Ck = apriori-gen(Lk-1);  //pembuatan
kandidat
                          //baru
   for all transactions t
  {
        C't = subset(Ck, t); //kandidat yang
                             //tampil pada t
        for all candidates c ∈ C't do
                c.count ++;
  }
Lk = { c ∈ C't | c.count ≥ minsup}
}
return ∪ kLk
```

Dimana :

L : *himpunan frequent itemset*

Minsup : *minimum support*

C : *himpunan kandidat itemset*

c : *kandidat itemset*

t : *transaksi*

2.4 Diagram Konteks

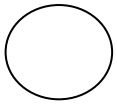
Diagram konteks bisa disebut dengan model sistem pokok (*fundamental system model*) mewakili keseluruhan elemen *software* dengan masukan (*input*) dan keluaran (*output*) yang diidentifikasi dengan anak panah masuk dan keluar memperlihatkan sumber data (Roger S Pressman, 1997). Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

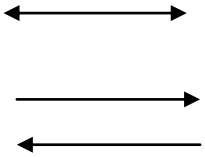
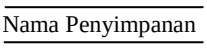
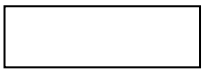
2.4.1 DFD *(Data Flow Diagram)*

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan proses kerja suatu sistem. Dalam DFD dijelaskan aliran masuk, aliran keluar, proses serta penyuntingan file yang digunakan

Ada empat komponen utama dalam DFD yaitu:

Tabel 2.1 Komponen DFD

Simbol	Keterangan
	**Proses**, menunjukkan transformasi dari masukan dan keluaran

Simbol	Keterangan
	**Aliran,** Komponen ini dipresentasikan dengan menggunakan panah yang menuju ke atau dari proses. Digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem dimana penyimpanan mewakili lokasi penyimpanan data
	**Penyimpanan,** komponen ini digunakan untuk memodelkan kumpulan data atau paket data.
	**Terminator,** komponen berikut ini dipresentasikan menggunakan persegi panjang, yang mewakili entity luar dimana sistem berkomunikasi.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Keuntungan menggunakan DFD adalah memudahkan pengguna yang kurang atau menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan atau dikembangkan.

2.5 Software Pendukung

Software pendukung merupakan alat yang yang digunakan untuk membangun sistem atau aplikasi yang akan digunakan. *Software* pendukung yang digunakan yaitu Visual Basic 6.0 dan Microsoft Access 2007

2.5 1 Visual Basic 6.0

Visual Basic adalah salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Jadi, dapat disimpulkan bahwa Visual Basic memiliki fungsi sebagai alat bantu untuk membuat berbagai macam program komputer. Dalam tugas akhir ini, Visual Basic digunakan sebagai *interface* program. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung

Object Oriented Programming (OOP). Bahasa pemrograman VB ini dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (Beginner's All-purpose Symbolic Instruction Code) yang dikembangkan pada era 1950-an. VB adalah salah suatu developement tools untuk membangun aplikasi dalam lingkungan Windows.

Pada pemrograman Visual, pengembangan aplikasi dimulai dengan pembentukkan *user interface*, kemudian mengatur properti dari objek-objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian-kejadian (event). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *Bottom Up*.

2.5.2 *Microsoft Acces*

Microsoft Access (atau ***Microsoft Office Access***) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan salah satu dari beberapa aplikasi *Microsoft Office*. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. *Microsoft Access* dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua kontainer basis data yang mendukung standar *ODBC*. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. *Access* juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.