BAB 2

TINJAUAN PUSTAKA

Pada tinjauan pustaka ini membahas tentang landasan teori yang medukung pembahasan yang berhubungan dengan sistem yang akan dibuat.

2.1 Data *Mining*

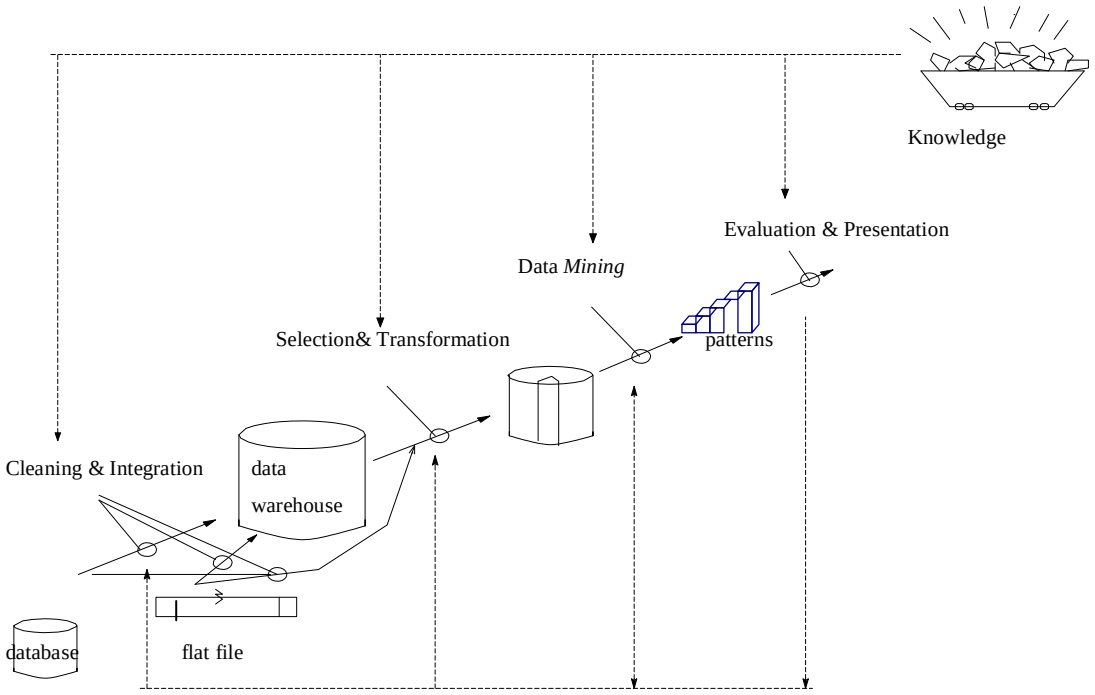
Data *mining* adalah kegiatan menemukan pola yang menarik dari data dalam jumlah besar dimana data tersebut dapat disimpan dalam bentuk *database*, data *warehouse*, atau penyimpanan informasi lainnya. Data *mining* berkaitan dengan bidang ilmu–ilmu lain, seperti *database system*, data *warehousing*, statistik, *machine learning*, *information retrieval*, dan komputasi tingkat tinggi. Selain itu, data *mining* didukung oleh ilmu lain seperti *neural netwo*rk, pengenalan pola, *spatial data analysis*, *image database*, *signal processing* (Han,*et al.*, 2006). Data *mining* adalah proses menganalisa data dari perspektif yang berbeda dan menyimpulkannya menjadi informasi-informasi penting yang dapat dipakai untuk meningkatkan keuntungan, memperkecil biaya pengeluaran, atau bahkan keduanya.

Data *mining* dapat disebut sebagai proses untuk menemukan korelasi atau pola dari ratusan atau ribuan *field* dari sebuah relasional *database* yang besar dan menggunakan teknologi pengenalan pola seperti yang terdapat pada teknik-teknik di statistika dan matematika (Larose,2005). Data *mining* juga disebut sebagai serangkaian proses untuk menemukan suatu pengetahuan atau informasi yang selama ini tidak diketahui dari data berskala besar dan sering juga disebut sebagai *knowledge discovery in database* (KDD) (Santosa,2007). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Data *mining* bukanlah suatu bidang yang sama sekali baru, data *mining* mewarisi banyak aspek dan teknik dari bidang ilmu

yang sudah mapan terlebih dahulu. Karakteristik data *mining*(Wirdasari& Ahmad.2011) sebagai berikut:

1. Data *mining*berguna untuk menemukan penemuan sesuatu yang pola data tertentu yang tersembunyi dan tidak diketahui sebelumnya.
2. Data *mining* biasa menggunakan data yang berukuran besar yang tersimpan dalam suatu basis data. Biasanya data yang besar digunakan data *mining* berguna untuk membuat keputusan yang kritis, terutama dalam strategi.

Berikut ini merupakan beberapa tahapan dalam data *mining* sebagai berikut:



Gambar 2.1 Tahapan Data *Mining*(Han, *et al.*, 2006)

1. Pembersihan Data (*DataCleaning*)

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 yang tidak relevan itu juga lebih baik dibuang karena

keberadaannya bisa mengurangi mutu atau akurasi dari hasil data *mining* nantinya. Pembersihan data juga akan mempengaruhi proses kerja dari teknik d*ata mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi Data (*DataIntegration*)

Data yang diperlukan untuk data *mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database*. 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 apabila terjadi 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. Transformasi juga dapat menentukan kualitas dari hasil data *mining* nantinya karena ada beberapa karakteristik dari teknik-teknik data *mining* tertentu yang tergantung pada tahapan ini.

5. Proses *Mining*

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

6. Evaluasi Pola (*Pattern Evaluation*)

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 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 dengan tahapan 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*.

2.2 Pengelompokan Data *mining*

Data *mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu :

1. Klasifikasi / *Classification*

   Klasisfikasi adalah proses menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri bisa berupa aturan “jika-maka”, berupa *decision tree*, formula matematis atau *neural network*. Proses klasifikasi biasanya dibagi menjadi dua fase yaitu *learning* dan test. Fase *learning*,sebagian data yang telah diketahui kelas datanya diumpankan

untuk membentuk model perkiraan. Kemudian pada fase test model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi model ini dapat dipakai untuk prediksi kelas data yang belum diketahui. Dalam klasifikasi, terdapat target variable kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori yaitu pendapatan tinggi, pendapatan sedang dan pendapatan rendah. Metode-metode klasifikasi adalah C4.5, *Bayesian, Neural network, Genetic algorithm, Fuzzy, Case-based reasoning,* dan*K-nearest neighbor.*

2. Klasterisasi / *Clustering*

Klasterisasi melakukan pengelompokan data tanpa berdasarkan kelas data tertentu, berbeda dengan asosiasi dan klasifikasi dimana kelas data telah ditentukan sebelumnya. Klasterisasi dapat dipakai untuk memberikan label pada kelas data yang belum diketahui. Prinsip dari klasterisasi adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas/cluster. Beberapa kategori algoritma clustering yang banyak dikenal adalah metode partisi dimana pemakai harus menentukan jumlah k partisi yang diinginkan lalu setiap data dites untuk dimasukkan pada salah satu partisi, metode lain yang telah lama dikenal adalah metode hierarki.

3. *Association Rule Mining*

Association Rule Mining adalah teknik *mining* untuk menemukan aturan asosiatif antara suatu kombinasi item. Analisis asosiasi dikenal juga sebagai salah satu teknik data *mining* yang menjadi dasar dari berbagai teknik data*mining* khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi. Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, yaitu *support* dan *confidence*. Algoritma yang paling populer dikenal sebagai Apriori yaitu pembuatan kandidat kombinasi item yang mungkin berdasar pada aturan tertentu lalu diuji apakah kombinasi item tersebut memenuhi syarat *support* minimum.

2.3 Aturan Asosiasi (*Assosiation Rule*)

Aturan Asosiasi adalah teknik untuk menemukan aturan asosiasi antara suatu kombinasi item. Aturan Asosiasi dikenal juga sebagai salah satu teknik data *mining* yang menjadi dasar dari berbagai teknikdata *mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi *(frequent pattern mining)* menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien.

Aturan Asosiasi adalah bentuk jika "kejadian sebelumnya" kemudian "konsekuensinya" (*if antecedent, then consequent)* yang diikuti dengan perhitungan aturan *support* dan *confidence*. Aturan Asosiasi ditentukan oleh dua parameter yaitu *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam database dan *confidence* (nilai kepastian) yaitu kuatnya hubungan antar item dalam aturan assosiatif. Kedua ukuran ini berguna dalam menentukan kekuatan sebuah pola dalam membandingkan pola tersebut dengan nilai minimum kedua parameter yang ditentukan oleh pengguna. Bila suatu pola memenuhi kedua nilai minimum parameter yang sudah ditentukan maka dapat diperoleh sebuah kesimpulan.

Aturan assosiatif biasanya dinyatakan dalam bentuk :
{Pinang Biji} →{Pakistan} (*support* = 40%, confidence = 50%)
Yang artinya : "50% dari transaksi di database yang memuat item Pinang Biji juga memuat item Pakistan.
Sedangkan 40% dari seluruh transaksi yang ada di database memuat kedua item itu."
Dapat juga diartikan : "Seorang pengguna jasa yang mengekspor pinang biji punya kemungkinan 50% untuk mengekspor ke negara Pakistan. Aturan ini cukup signifikan karena mewakili 40% dari catatan transaksi selama ini."

2.4 Metodologi Dasar Analisis Aturan Asosiasi

a. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam database. Nilai*support* sebuah item diperoleh dengan rumus (1) :

$$Support\ (A) = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Total Transaksi}} \quad \text{.....(1)}$$

Pada rumus menjelaskan bahwa nilai *support* diperoleh dengan cara mencari jumlah transaksi yang mengandung nilai A (satu item) dibagi dengan jumlah keseluruhan transaksi dan untuk nilai *support* dari 2 item diperoleh dari rumus (2) :

$$Support\ (A \cap B) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}} \quad \text{.....(2)}$$

Pada rumus menjelaskan bahwa nilai *support* diperoleh dengan cara mencari jumlah transaksi yang mengandung nilai A dan B (item pertama bersamaan dengan item yang lain) dibagi dengan jumlah keseluruhan transaksi.

b. Pembentukan aturan asosiasi

Setelahpola frekuensi tinggi ditemukan, barulah dicari aturan assosiasi yang memenuhi syaratminimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A $\rightarrow$B. Nilai *confidence* dari aturan A $\rightarrow$B diperoleh dari rumus (3):

$$\text{Confidence} = P(B \mid A) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}} \quad \text{.....(3)}$$

Pada rumus menjelaskan bahwa nilai *confidence* diperoleh dengan cara mencari jumlah transaksi yang mengandung nilai A dan B (item pertama bersamaan dengan item yang lain) dibagi dengan jumlah transaksi yang mengandung A (item pertama).

2.5 Algoritma Apriori

Algoritma Apriori adalah salah satu algoritma yang melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule.* Algoritma Apriori menggunakan pengetahuan frekuensi atribut yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Pada algoritma Apriori menentukan kandidat yang mungkin muncul dengan cara memperhatikan minimum *support* dan minimum *confidence*. *Support* adalah nilai pengunjung atau persentase kombinasi sebuah item dalam *database*.

Untuk penerapan algoritma Apriori, secara umum dibutuhkan struktur data untuk menyimpan *candidate frequentitemset* untuk suatu iterasi ke *k* dan untuk menyimpan *frequent itemset* yang dihasilkan. Ketika membaca tiap itemdari seluruhtransaksi, selain mendapatkan item-itembaru juga dilakukan perhitungan nilai *support*item-itemyang sudah ditemukan,sehingga untuk mendapatkan *candidate 1*-item*set* beserta nilai *support*-nya cukup membutuhkan satu kali pembacaandata. Di iterasi pertama ini, *support* dari setiap item dihitung dengan men-scan database. Setelah *support* dari setiap item didapat, item yang memiliki *support* diatas minimum *support* dipilih sebagai pola frekuensi tinggi dengan panjang 1 atau sering disingkat 1-itemset. Iterasi kedua menghasilkan 2-itemset yang tiap set-nya memiliki dua item. Pertama dibuat kandidat 2-itemset dari kombinasi semua 1-itemset. Lalu untuk tiap kandidat 2-itemset ini dihitung *support*-nya dengan men-scan database. *Support* disini artinya jumlah transaksi dalam database yang mengandung kedua item dalam kandidat 2-itemset. Setelah *support* dari semua kandidat 2-itemset didapatkan, kandidat 2-itemset yang memenuhi syarat minimum *support* dapat ditetapkan sebagai 2-itemset yang juga merupakan pola frekuensi tinggi dengan panjang 2. Demikianjuga pada iterasi ke-k, dimanakandidat k-itemset dibentuk dari kombinasi (k-1)-itemset yang didapat dari iterasi sebelumnya. Hal ini merupakan salah satu ciri dari algoritma Apriori yaitu adanya pemangkasan kandidat k-itemset yang subset-nya yang berisi k-1 item tidak termasuk dalam pola frekuensi tinggi dengan panjang k-1.

2.6 Algoritma CT-Pro

Algoritma CT-Pro merupakan salah satu algoritma pengembangan dari *FP-Growth*. Perbedaannya terdapat pada langkah kedua dimana *FP-Growth* membuat *FP-Tree* sedangkan CT-Promembuat *Compressed FP-Tree* (*CFP- Tree*). Pada tahap *mining* algoritma CT-Pro juga menggunakan pendekatan *bottom-up* dimana item pada item *tabel* dan *CFP-Tree* dilakukan *scan* dari jumlah terkecil hingga terbesar.

Algoritma CT-Pro memiliki tiga tahap yaitu:

1. Menemukan item-item yang *frequent*
2. Membuat struktur data *CFP-Tree*
3. Melakukan *mining frequent patterns*

Langkah-langkah kerja algoritma CT-Pro:

1. Mencari *Frequent* item, pada tahap ini terjadi proses-proses sebagai berikut:
   - Dari *dataset* yang ada, dilakukan seleksi berdasarkan *minimum support* yang ditentukan sehingga menghasilkan *frequent* item lalu dihitung frekuensi kemunculan setiap item sehingga menghasilkan *GlobalItem tabel*.
2. Membangun *CFP-Tree*, pada tahap ini terjadi proses-proses sebagai berikut:
   - *Mengurutkan frequent* item berdasarkan *GlobalItem tabel* yang ada secara menurun (diurutkan mulai dari item berfrekuensi terbesar hingga terkecil) dan dibentuk *Global CFP-Tree*
3. *Mining*, pada tahap ini terjadi proses-proses sebagai berikut:
   - Pada tahap *mining* ini, algoritma CT-Pro bekerja dengan melakukan bottom -up *mining* sehingga *Global Item tabel* diurutkan mulai dari item berfrekuensi terkecil hingga terbesar.
   - Untuk setiap item yang terdaftar pada *GlobalItem tabel* yang telah diurutkan, dilakukan pencarian *node* yang berkaitan dengan item tersebut pada *Global CFP-Tree* yang kemudian disebut sebagai *Local Frequent item* dan digunakan untuk membuat *Local Item Tabel.*
   - Setelah itu, dibuat *Local CFP-Tree* berdasarkan *Local Item Tabel* yang terbentuk. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *GlobalItemtabel* yang terbentuk

dari *GlobalItem tabel* data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah Local Item tabel yang terbentuk dari Local *Frequent item.*

- Dari *Local CFP-Tree* dibentuk *frequent pattern* sesuai dengan item yang di*mining*.

Dari *frequent pattern* dihitung masing-masing item yang memenuhi dihitung *confidence*nya. Apabila memenuhi minimum *confidence* maka masing-masing item yang bersangkutan dijadikan sebagai *knowledge*.

2.7 Penelitian Terdahulu

Penelitian lain yang menggunakan algoritma Apriori ataupun algoritma CT-Pro yaitu penelitian yang telah dilakukan Aritonang (2012) melakukan penelitian mengenai pengambilan keputusan untuk korelasi pembelian produk yang menggunakan algoritma apriori yang bertujuan untuk mengetahui data yang sering muncul dan mengetahui aturan tata letak produk.

Penelitian yang telah dilakukan Ruldeviyani dan Fahrian (2008), yang mengimplementasikan algoritma *association rules* yaitu Apriori, FP-Growth, CT-Pro, dan Apriori Cristian Borgelt sebagai bagian dari pengembangan *datamining workbench*. Dari hasil pengujian yang dilakukan pada *dataset chess,*CT-Pro paling cepat dibandingkan dengan algoritma yang lain. Keunggulan CT-Pro adalah dari penggunaan memori yang lebih hemat dan digunakannya struktur data *CFP-Tree* yang memungkinkan proses pencarian *frequent* item*set* menjadi lebih cepat. Pada *support*80, algoritma FP-Growth lebih lambat jika dibandingkan dengan Apriori Christian Borgelt karena FP-Growth mengalami pertambahan ukuran *FP-Tree* di memori.

Dhivya and Kalpana (2010), dalam penelitian ini yang ditunjukkan dalam kurva kinerja kecepatan eksekusi, jelas bahwa CT-Pro melakukan lebih baik daripada semua algoritma dalam segala situasi. Algoritma CT-Apriori dan CT-Pro melebihi dari algoritma dasarnya yaitu algoritma Apriori dan FP-Growth. Kesenjangan kinerja antara CT-Apriori dan CT-Pro lebih menonjol di ambang batas yang lebih rendah.