BAB II
TINJAUAN PUSTAKA

2.1 *Data Mining*

2.1.1 Definisi*Data Mining*

Data Mining adalah proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar. Data mining juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan [3].

Menurut [4], data mining merupakan integrasi dari beberapa disiplin ilmu seperti teknologi database dan data warehouse, statistik, *machine learning*, pengenalan/pencocokan pola, *neural networks*,visualisasi data, *information retrieval*,pemrosesan sinyal dan gambar,dan spasial atau temporal analisis data.

Dari pemrosesan data mining kita dapat melihat, melihat pola data yang tersimpan pada database dari beberapa sudut dan dengan adanya informasi tersebut dapat diterapkan menjadi pendukung keputusan, kontrol proses dan menejemen informasi.

Fungsionalitas data mining digunakan untuk menentukan pola yang terdapat di dalamnya. Pada umumnya sifat data mining dibagi menjadi dua yaitu prediktif dan deskriptif. Prediktif pada umumnya dilakukan untuk memprediksi sesuatu berdasarkan data yang ada. Deskriptif merupakan proses data mining yang mengkarakterkan berdasarkan sifat data pada database.

2.1.2 Pengelompokan *Data Mining*

Dalam [2] *Data Mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

a. Deskripsi

   Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan baris data (*record*) lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

c. Prediksi

Prediksi hampir sama 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 keadaan yang tepat) untuk prediksi.

d. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

e. Pengklasteran (*Clustering*)

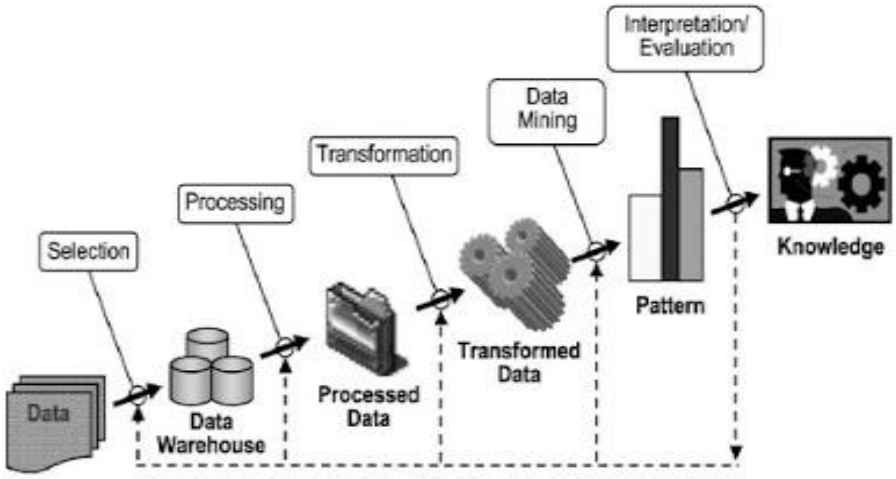
Pengklasteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas obyek-obyek yang memiliki kemiripan. Klaster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan *record* dalam klaster yang lain. Berbeda dengan klasifikasi, pada pengklasteran tidak ada variabel target. Pengklasteran tidak melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

f. Asosiasi

Tugas asosiasi dalam *Data Mining* adalah untuk menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja*(Market Basket Analysis)*.

2.2 Tahapan-tahapan *Data Mining*

Tahapan yang dilakukan pada proses *Data Mining* diawali dari seleksi data dari data sumber ke data target, tahap *preprocessing* untuk memperbaiki kualitas data, transformasi, *Data Mining* serta tahap interpretasi dan evaluasi yang menghasilkan output berupa pengetahuan baru yang diharapkan memberikan kontribusi yang lebih baik. Secara *detail* dijelaskan sebagai berikut [2]:



Gambar 2.1 Tahapan *Data Mining*

1. *Data selection*

   Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang digunakan untuk proses *Data Mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing / cleaning*

   Sebelum proses *Data Mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.

3. *Transformation*

   Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *Data Mining.* Proses *coding*dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

   Data Mining adalah 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.

5. *Interpretation / evaluation*

   Pola informasi yang dihasilkan dari proses *Data Mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut interpretation. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.3 Binerisasi dan Diskretisasi

Biasanya algoritma analisis pola asosiasi membutuhkan data dalam bentuk atribut yang nilainya biner. Transformasi data dari tipe kontinyu dan diskret ke atribut biner disebut binerisasi, sedangkan transformasi data dari atribut kontinyu ke atribut kategoris disebut diskretisasi. Proses binerisasi dan diskretisasi menjadi penting karena hasil yang baik dari proses ini akan berpengaruh pada hasil kinerja algoritma *data mining.*

Analisis asosiasi membutuhkan data dengan atribut biner yang asimetris karena dalam analisis asosiasi hanya ada atribut dengan nilai 1 yang dianggap penting [3].

2.4 *Association rule*

2.4.1 Pengertian *Association rule*

Association rule adalah suatu prosedur yang mencari hubungan atau relasi antara satu *item* dengan *item* lainnya. *Association rule* biasanya menggunakan "*if*" dan "*then*" misalnya "*if* A *then* B *and* C", hal ini menunjukkan jika A maka B dan C. Dalam menentukan *association rule* perlu ditentukan *support* dan *confidence* untuk membatasi apakah *rule* tersebut *interesting* atau tidak [4].

Association rule berguna untuk menemukan hubungan penting antar *item* dalam setiap transaksi, hubungan tersebut dapat men*and*akan kuat tidaknya suatu aturan dalam asosiasi, Tujuan *association rule* adalah untuk menemukan keteraturan dalam data. *Association rule* dapat digunakan untuk mengidentifikasi *item-item* produk yang mungkin dibeli secara bersamaan dengan produk lain, atau dilihat secara bersamaan saat mencari informasi mengenai produk tertentu. Dalam pencarian *association rule*, diperlukan suatu variabel ukuran kepercayaan *(interestingness measure)* yang dapat ditentukan oleh *user*, untuk mengatur batasan sejauh mana dan sebanyak apa hasil *output* yang diinginkan oleh *user*.

2.4.2 Ukuran Kepercayaan *Rule(Interestingness Measure)*

Menurut [4] terdapat dua ukuran kepercayaan yang menunjukkan kepastian dan tingkat kegunaan suatu *rule* yang ditemukan yaitu:

1. *Support*

   Support (dukungan) merupakan suatu ukuran yang menunjukkan seberapa besar dominasi suatu *item* atau *itemset* dari keseluruhan transaksi.

2. *Confidence*

   Confidence (tingkat kepercayaan) adalah suatu ukuran yang menunjukkan hubungan antar *item* secara conditional (misalnya seberapa sering *item* B dibeli jika orang membeli *item* A).

   Untuk menemukan aturan asosiasi seperti yang diharapkan maka harus menemukan nilai dari *support* yang telah ditentukan. *Support* tersebut merupakan jumlah *item* pada setiap transaksi yang ada didalam *database*.

   Untuk dapat menemukan nilai *support* kita dapat mencari semua aturan yang jumlah *support* ≥ *minimum support*. Dalam hal ini dapat digunakan sebagai cara untuk menemukan sebuah nilai *confidence*. Nilai *confidence* ditentukan dari nilai *support* suatu aturan dalam sebuah transaksi.

   Jika *itemset* pada setiap transaksi tidak sering muncul (*infrequent*), maka kandidat yang tidak sesuai dengan nilai *support* ≥ *minimum support* tersebut harus segera dipangkas tanpa harus menghitung *confidence*nya. Strategi

umum digunakan oleh banyak algoritma penggalian aturan asosiasi adalah memecahkan masalah ke dalam dua pekerjaan utama, yaitu:

1. *FrequentItemsetGeneration*

   Tujuannya adalah mencari semua *itemset* yang memenuhi ambang batas *minimum support*. *Itemset* itu disebut *itemset frequent*(*Itemset* yang sering muncul)

2. *Rules Generation*

   Tujuannya adalah mengekstrak aturan dengan *confidence* tinggi dari *itemsetfrequent* yang ditemukan dalam langkah sebelumnya. Aturan ini kemudian disebut aturan yang kuat (*Strong rules*) [3].

2.5 Algoritma Apriori

Pada bagian ini akan dijelaskan tentang algoritma Apriori sebagai metode yang digunakan dalam tugas akhir ini, yang meliputi definisi, langkah-langkah dan contoh kasus dengan menggunakan algoritma Apriori.

2.5.1 Pengertian Algoritma Apriori

Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian *frequentitemset* dengan menggunakan teknik *association rule*. Algoritma Apriori menggunakan *knowledge* mengenai *frequentitemset* yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma Apriori untuk menentukan kandidat-kandidat yang mungkin muncul dengan cara memperhatikan *minimumsupport*[2].

Algoritma apriori termasuk jenis aturan asosiasi pada *Data Mining*. Selain algoritma 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*. Analisis asosiasi atau *association rulemining* adalah teknik *Data Mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap :

1. Analisis pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat *minimum* dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut : [2]

$$Support(A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ transaksi} \quad \text{... (2.1)}$$

Gambar 2.2 Rumus Nilai *Support*1 *item*

Nilai *support* dari 2 *item* diperoleh dengan menggunakan rumus:

$$Support(A,B) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum transaksi} \quad \text{... (2.2)}$$

Gambar 2.3 Rumus Nilai *Support*2 *item*

Frequentitemset menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai *minimum* yang ditentukan (ϕ). Misalkan ϕ = 2, maka semua *itemset*s yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent*. Himpunan dari *frequent*k-*itemset* dilambangkan dengan F_k.

2. Pembentukan Aturan Asosiasi

Setelah semua pola pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat *minimum* untuk *confidence* dengan menghitung *confidence* aturan asosiatif A → B.Nilai *Confidence* dari aturan A → B diperoleh rumus berikut.

$$Confidence = P(B|A) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi\ mengandung\ A} \quad \text{... (2.3)}$$

Gambar 2.4 Rumus Nilai *Confidence*

Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan *Support* × *Confidence*. Aturan diambil sebanyak n-aturan yang memiliki hasil terbesar.

2.5.2 Proses Utama Algoritma Apriori

Proses Utama Algoritma Apriori untuk meningkatkan efisiensi dari pencarian k-*itemset*, dapat digunakan suatu metode tambahan yang dinamakan Apriori *Property*. Metode ini dapat mengurangi lingkup pencarian sehingga waktu pencarian dapat dipersingkat.

Menurut [4] terdapat dua proses utama yang dilakukan dalam algoritma Apriori, yaitu:

1. *Join* (Penggabungan).

   Pada proses ini setiap *item* dikombinasikan dengan *item* yang lainnya sampai tidak terbentuk kombinasi lagi. Untuk menemukan Lk, suatu set dari kandidat k-*itemset* dihasilkan dengan cara men-*join*kan Lk-1 dengan dirinya sendiri. Set kandidat hasil *join* ini nanti akan dinotasikan sebagai Ck. Adapun aturan dari *join* ini adalah setiap kandidat yang dihasilkan tidak boleh meng*and*ung kandidat yang kembar antara satu dengan yang lainnya.
2. *Prune* (Pemangkasan).

   Pada proses ini, hasil dari *item* yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan *minimumsupport* yang telah ditentukan oleh *user*. Semua (k-1)-*itemset* yang tidak *frequent* tidak mungkin dapat menjadi subset dari *frequent* k-*itemset*. Oleh karena itu, jika ada (k-1) subset dari kandidat k-*itemset* yang tidak termasuk dalam Lk-1, maka kandidat tidak mungkin *frequent* juga dan oleh karena itu dapat dihapus dari Ck.

2.5.3 Langkah-langkah dari proses Algoritma Apriori

Langkah-langkah algoritma Apriori untuk mendapatkan *rule*s yang diinginkan oleh *user*, antara lain:

1. Melakukan *scandatabase* untuk mendapat kandidat 1-*itemset*, yaitu C1 (Himpunan *item* yang terdiri dari 1 *item*) dan menghitung nilai *support*nya. Bandingkan nilai *support* dengan *minimumsupport* yang sudah ditentukan, jika nilainya lebih besar atau sama dengan *minimumsupport*, maka *itemset* tersebut termasuk dalam *large itemset* yaituL1 (*Large itemset* dengan 1 *itemset*)

2. *Itemset* yang tidak termasuk dalam *large itemset* tidak disertakan dalam iterasi selanjutnya (dilakukan *pruning*).
3. Himpunan L1 hasil iterasi pertama akan digunakan untuk iterasi selanjutnya. Pada L1 dilakukan proses *join* terhadap dirinya sendiri untuk membentuk kandidat 2 *itemset* (C2). Bandingkan lagi *support* dari *item-item* C2 dengan *minimumsupport*, bila tidak kurang dari *minimumsupport*, maka *itemset* tersebut masuk dalam *large itemset* L2. Pada iterasi selanjutnya, hasil *large itemset* pada iterasi sebelumnya (Lk-1) akan dilakukan proses *join* terhadap dirinya sendiri untuk membentuk kandidat baru (Ck), dan *large itemset* baru (Lk). Setelahnya dilakukan proses *pruning* pada *itemset* yang tidak termasuk dalam Lk-.
4. Dari seluruh *large itemset* yang memenuhi *minimumsupport* (*frequentitemset*) dibentuk *association rule* dan nilai *confidence*nya. Aturan-aturan yang nilai *confidence*nya lebih kecil dari *minimumconfidence*, tidak termasuk dalam *association rule* yang dipakai.

2.6 *Lift Ratio*

Lift Ratio adalah parameter penting selain *support* dan *confidence* dalam *association rule*. *Lift ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. *Lift ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar *item* A dibeli bersamaan dengan *item* B. *Lift ratio* dapat dihitung dengan rumus [5] :

$$Nilai\ Lift = \frac{Support\ \ (A \cap B)}{Support\ \ (A)\ x\ Support\ \ (B)} \qquad \ldots (2.4)$$

Sebuah transaksi dikatakan valid jika mempunyai nilai lift/improvement lebih dari 1, yang berarti bahwa dalam transaksi tersebut *item* A dan *item* B benar-benar dibeli secara bersamaan.

2.7 Penelitian Sebelumnya

Beberapa riset yang telah dilakukan berkaitan dengan kasus asosiasi yang menggunakan metode apriori antara lain:

Penelitian yang berjudul *"IMPLEMENTASI DATA MINING PADA PENJUALAN PRODUK ELEKTRONIK DENGAN ALGORITMA APRIORI (STUDI KASUS :KREDITPLUS)"*. Penelitian yang dilakukan pada salah satu perusahaan pembiayaanberagam produk elektronik ini dilaksanakan oleh Dewi Kartika Pane. Penelitian ini bertujuan untuk mengetahui sejauh mana algoritma apriori dapat membantu pengembangan strategi pemasaran pada perusahaan tersebut. data yang digunakan berjumlah 13 item merk laptop. Tidak disebutkan berapa banyak transaksi yang diproses, hanya saja penelitian ini dilaksanakan menggunakan data transaksi mulai april 2012 sampai maret 2013. Proses yang dilakukan menggunakan tools Tanagra versi 1,4 dan menghasilkan 2 aturan dengan minimal *support* = 30% dan minimal *confidence* = 60%. Hasil penelitian disebutkan bahwa penjualan tebanyak didapatkan oleh produk merk acer dan toshiba. Peneliti juga menyebutkan bahwa perusahaan dianjurkan untuk menambah persediaan produk laptop dengan merk toshiba dan acer, selain itu juga menyebutkan menganjurkan perusahaan untuk menyusun strategi pemasaran pada merk lain.

Penelitian selanjutnya oleh Hernawati yang berjudul *"APLIKASI DATA MINING UNTUK PERMODELAN PEMBELIAN BARANG DENGAN MENGGUNAKAN ALGORITMA APRIORI"* yang dilaksanakan oleh Almon Junior Simanjuntak. Penelitian tersebut menggunakan 3 item barang yang dapat menghasilkan aturan-aturan asosiatif antar barang. Penelitian tersebut juga mengungkapkan bahwa semakin banyak data yang diproses maka semakin besar pula alokasi memori untuk menjalakan proses. Hal ini disebabkan karena proses yang digunakan pada algoritma apriori melakukan pemindaian ulang secara terus menerus pada tiap iterasinya. Semakin banyak jumlah kombinasi item, maka nilai support dan confidencenya semakin kecil. Hasil akhir dari penerapan algoritma apriori dapat digunakan untuk mengetahui gambaran umum kebiasaan belanja

konsumen sehingga pengusaha dapat menentukan stok barang apa saja yang perlu diperbanyak dan menentukan tata letak barang berdasarkan kelompok yang palingsering dibeli konsumen.

Penelitian selanjutnya *"PENGGUNAAN METODE APRIORI UNTUK ANALISA KERANJANG PASAR PADA DATA TRANSAKSI PENJUALAN MINIMARKET MENGGUNAKAN JAVA & MYSQL"*. Penelitian tersebut dilaksanakan oleh Devi Dinda Setiawati Universitas Gunadarma. Pada penelitian tersebut ditujukan untuk mengetahui strategi pemasaran pada sebuah minimartket dan diharapkan dapat membantu untuk mengetahui kebiasaan berbelanja konsumen. Ujicoba yang dilakukan pada 55 macam produk dan data transaksi satu hari sejumlah 100 transaksi. Dari percobaan yang telah dilakukan, didapatkan 8 aturan asosiasi data dengan parameter minimal *confidence* > = 10% dan minimal *support* > = 5%. Waktu yang dibutuhkan untuk menyelesaikan proses tersebut tidak dicantumkan, akan tetapi dianggap sesuai dengan jumlah data yang diproses.