BAB II

DASAR TEORI

Pada bab ini membahas mengenai teori-teori dan referensi yang digunakan dalam penyelesaian skripsi ini.

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Yanxi Liu yang berjudul "*Study on Application of Apriori Algorithm in Data Mining"* tahun 2010. Penelitian ini menekankan pada pengaplikasian teknologi *data mining* kedalam manajemen stok perusahaan. Dengan kumpulan data yang besar akan digali informasi-informasi dan aturan yang tersembunyi didalamnya. Hasil dari penelitian ini berupa kesimpulan bahwa untuk proses *scanning database* dilakukan berulang-ulang. Jika terdapat *frequent item set* yang besarnya 10, maka proses *scanning* data juga dilakukan sebanyak 10 kali. Untuk proses mencari kandidat *itemset* yang besar, diperlukan waktu komputasi dan alokasi yang besar pula. Jika *minimum support* yang ditetapkan terlalu tinggi, maka data yang dihasilkan akan berkurang, sehingga kemungkinan tidak ditemukan aturan, begitu juga sebaliknya. Hal ini akan mempersulit pembuatan keputusan[LIU-10].

Penelitian yang dilakukan oleh Bete Noranita dan Nurdin Bahtiar yang berjudul "Implementasi *Data Mining* Untuk Menemukan Pola Hubungan Tingkat Kelulusan Mahasiswa Dengan Data Induk Mahasiswa" tahun 2010. Dalam menemukan pola yang dicari penelitian ini menggunakan algoritma Apriori. Penggalian data yang dilakukan yaitu pencarian hubungan antara data induk mahasiswa dengan proses masuk, asal sekolah, kota asal sekolah, dan program studi. Dalam penelitian ini dihasilkan kesimpulan bahwa dalam pencarian hubungan tingkat kelulusan dengan data induk mahasiswa diukur oleh nilai *support* dan *confidence* antar item [NOR-10].

Penelitian yang dilakukan oleh Erwin yang berjudul "Analisis *Market Basket* Dengan Algoritma *Apriori* dan *FP-Growth*" pada tahun 2009. Penelitian ini dilakukan untuk membandingkan kinerja antara algoritma *Apriori* dengan *FP-Growth*. Dengan menggunakan parameter yang sama yaitu *minimum support* dan *minimum confidence* menghasilkan kesimpulan bahwa algoritma *Apriori* membutuhkan waktu komputasi yang lebih lama dibandingkan algoritma *FP-Growth* dalam menghasilkan *frequent itemsets*. Algoritma *Apriori* berulang kali melakukan pemindaian data dan juga membutuhkan alokasi memori yang lebih besar dalam pencarian *itemsets*. Sedangkan algoritma *FP-Growth*, hanya membutuhkan dua kali *scanning database* dalam pencarian *frequent itemsets* sehingga waktu yang dibutuhkan juga relatif lebih singkat [ERW-09].

2.2 Data Mahasiswa

Data yang akan dipakai pada skripsi ini merupakan data mahasiswa Fakultas Ekonomi. Data mahasiswa yang dipakai meliputi asal propinsi, NIM (Nomor Induk Mahasiswa), nama mahasiswa, angkatan, seleksi (jalur masuk), IPK lulus dan juga lama studi. Fakultas ekonomi memiliki beberapa jurusan, yaitu Jurusan Ekonomi Pembangunan (Ekonomi Islam dan Keuangan dan Perbankan), Jurusan Manajemen, Jurusan Akuntansi [TIM-12].

2.1.1 Macam-macam Jalur Masuk Mahasiwa Baru

Pada program sarjana yang ada di Universitas Brawijaya, terdapat beberapa jalur masuk yang bisa dipilih oleh calon mahasiswa baru, antara lain [BIR-09] :

a. Penjaringan Siswa Berprestasi (PSB)

   Penjaringan ini dilakukan tanpa ujian tulis (tes), dimaksudkan untuk menjaring calon mahasiswa yang berprestasi, baik dibidang akademik maupun non akademik.

b. Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN)

   Seleksi ini dilakukan melalui ujian tulis dan dilaksanakan secara nasional, bersama-sama seluruh Perguruan Tinggi Negeri di Indonesia.

c. Seleksi Program Minat dan Kemampuan (SPMK)

   Seleksi ini dilakukan melalui ujian tulis secara mandiri oleh Universitas Brawijaya bagi mahasiswa yang berminat dan mempunyai kemampuan.

d. Seleksi Program Kemitraan Sekolah (SPKS)

Seleksi ini dilakukan melalui ujian tulis maupun tanpa ujian tulis berdasarkan kemitraan dengan sekolah, dimaksudkan untuk menjaring calon mahasiswa yang berprestasi di bidang akademik.

e. Seleksi Program Kemitraan Instansi (SPKIns)
   Seleksi ini dilakukan melalui ujian tulis berdasarkan kemitraan dengan instansi.
f. Seleksi Program Kemitraan Daerah (SPKD)
   Seleksi ini dilakukan melaui ujian tulis berdasarkan kemitraan dengan Pemerintah Daerah.
g. Seleksi Program Internasional (SPI)
   Seleksi ini dilakukan melalui ujian tulis berdasarkan kemitraan dengan pihak luar negeri.
h. Seleksi Alih Program (SAP)
   Seleksi ini dilakukan melalui ujian tulis bagi lulusan program diploma dan perguruan tinggi yang setara.
i. UB IV
   Merupakan seleksi penerimaan mahasiswa baru kampus IV UB. Kampus IV UB sendiri terletak di kota Kediri, Jawa Timur.
j. Exchange Program
   Merupakan program pertukaran mahasiswa.
k. Seleksi Program Khusus Penyandang Disabilitas (SPKPD)
   Seleksi yang diperuntukan bagi pengandang disabilitas yang mempunyai kapabilitas dalam bidang keilmuan dari Jurusan atau Program Studi yang akan dipilih. Penerimaan mahasiswa baru ini melalui seleksi administrative, tes wawancara dan psikotes yang dilakukan oleh panitia penerimaan mahasiswa baru UB.

2.1.2 Predikat Kelulusan

Kelulusan seorang mahasiswa dibagi menjadi beberapa predikat kelulusan. Predikat kelulusan ini ditentukan oleh besarnya indeks prestasi (IP), predikat ini dibagi menjadi 3, yaitu Memuaskan, Sangat Memuaskan, dan Dengan Pujian yang dinyatakan pada transkrip akademik. Pengaturan IPK sebagai dasar predikat kelulusan, antara lain:

- IPK 2.00 – 2.75 Memuaskan
- IPK 2.76 – 3.50 Sangat Memuaskan
- IPK 3.51 – 4.00 Dengan Pujian (*cumlaude*)

Untuk predikat Dengan Pujian (*cumlaude*) ditentukan juga dengan lama studi yaitu untuk gelar sarjana maksimal 5 tahun atau 10 semester [TIM -12].

2.3 Data Mining

Data mining memiliki beberapa pandangan, seperti *knowledge* ataupun *pattern recognition*. Kedua istilah tersebut sebenarnya memiliki ketepatannya masing-masing. Istilah *knowledge discovery* atau penemuan pengetahuan tepat digunakan karena tujuan utama dari data mining memang untuk mendapatkan pengetahuan yang masih tersembuyi didalam bongkahan data. Istilah *pattern recognition* atau pengenalan pola pun tepat untuk digunakan karena pengetahuan yang hendak digali berbentuk pola-pola yang mungkin juga masih perlu digali dari dalam bongkahan data yang tengah dihadapi [SUS-10].

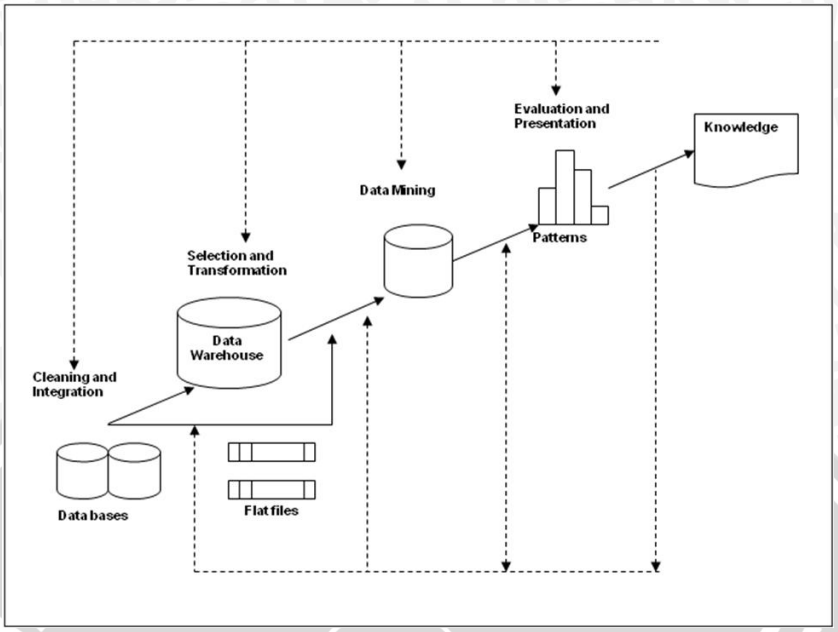
2.2.1 Pengertian Data Mining

Data mining merupakan solusi yang mampu menemukan kandungan informasi yang tersembunyi berupa pola dan aturan dari sekumpulan data yang besar [HAN-06].

Data mining adalah analisis dari pengamatan data set untuk menemukan hubungan yang tak terduga dan untuk meringakas data dengan cara baru yang lebih mudah dimengerti dan lebih berguna bagi pemilik data [HAN-01].

2.2.2 Tahap-tahap Data Mining

Tahap-tahap dalam data mining dapat dilihat dalam gambar yang terdiri dari (Han, 2012):



Gambar 2.1 *Data mining* sebagai sebuah tahap dalam proses KDD

Sumber: [HAN-06]

1. *Data Cleaning* (Pembersihan Data)

   Pada tahap ini mempunyai tujuan untuk membuang noise dan observasi duplikat (menhapus data yang yang tidak perlu dan tidak konsisten).

2. *Data Integration* (Integrasi Data)

   Pada tahap ini merupakan proses untuk melakukan penambahan data yang sudah ada dengan data-data yang akan digunakan (data relavan) dan digunakan oleh KDD (*Knowledge Discovery in Database*).

3. *Data Selection*

   Pada tahap ini dilakukan proses pemilihan data yang relevan untuk dilakukan analisis dari *database*.

4. *Data Transformation*

   Pada tahap ini dilakukan proses transformasi dan konsolidasi data kedalam bentuk tertentu untuk penggalian oleh ringkasan performa atau operasi agregasi .

5. *Data Mining*

   Data mining merupakan sebuat proses dimana kemampuan dari metode yang diaplikasikan untuk menemukan pola atau informasi yang menarik.

6. *Pattern Evaluation*

Proses ini bertujuan untuk mengidentifikasi yang menampilkan basis pengetahuan yang benar-benar menarik pada hasil *data mining*.

7. *Knowledge Presentation*

   Pada tahap ini merupakan visualisasi dan teknik penampilan pengatahuan yang digunakan untuk memudahkan *ussr* (pengguna) guna mengerti terhadap pola informasi yang dihasilkan.

2.4 *Association Rules*

Aturan asosiasi (*association rules*) merupakan teknik data mining untuk menemukan aturan asosiatif antara kombinasi item. Analisa asosiasi ini menjadi dikenal banyak kalangan karena sering digunakan menganalisa keranjang belanja swalayan (*market basket analysis*).

Analisis asosiasi dipakai sebagai acuan dalam data mining. Salah satu tahap dari analisis asosiasi yaitu analisis pola frekuensi tinggi (*frequent pattern mining*) menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Penting tidaknya suatu aturan asosiatif dapat diketahui dengan tiga parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian), dan nilai *Lift* (nilai kekuatan). *Support* adalah persentase kombinasi item tersebut dalam *database*, *confidence* adalah kuatnya hubungan antar item dalam aturan asosiatif. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap, yaitu :

a. Analisa pola frekuensi tinggi

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

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

Sumber : [GUN-12]

Sedangkan nilai *support* dari 2 item diperoleh dari rumus berikut :

$$Support(A \cap B) = \frac{\text{Jumlah transaksi mengandung A dan B}}{\text{Total transaksi}} \quad \text{.... (2.2)}$$

Sumber : [GUN-12]

b. Pembentukan aturan asosiatif

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

$$Confidence = P(B \mid A) = \frac{\text{Jumlah transaksi mengandung A dan B}}{\text{Jumlah transaksi mengandung A}}$$

(2.3)

Sumber : [GUN-12]

2.5 *Lift Ratio*

Lift Ratio merupakan ukuran dari tingkat kekuatan pola yang dihasilkan. Pola yang mempunyai nilai lift rasio lebih besar dari 1, menunjukkan bahwa pola tersebut memiliki kekuatan dalam polanya. Semakin besar nilai lift rasio suatu pola, semakin besar pula tingkat kekuatan pola tersebut. Sedangkan untuk pola yang mempunyai nilai lift rasio kurang dari satu, kekuatan pola tersebut dianggap lemah. Nilai lift rasio didapatkan dari perhitungan antara nilai *confidence* dengan nilai *benchmark confidence* (*confidence'*).

$$\text{Bencmark Confidence} = \frac{\text{Jumlah kemunculan item dalam consequent}}{\text{Total transaksi dalam database}} \quad ...(2.4)$$

Sumber : [AMI-10]

$$\text{Lift Ratio} = \frac{\text{Confidence}}{\text{Bencmark Confidence}} \quad ....(2.5)$$

Sumber : [AMI-10]

2.6 Algoritma *FP-Growth*

Frequent Pattern Growth (FP-Growth) adalah algoritma yang mengadopsi *devide* dan *conquer* dalam proses mendapatkan *frequent itemset*. *Devide* merupakan tahap pembagian permasalahan yang besar menjadi permasalahan yang lebih kecil. Sedangkan *conquer* merupakan tahap pembagian secara terus-menerus sampai didapatkan bagian maasalah kecil yang lebih mudah untuk dipecah. Dalam penemuan *frequent itemset*, algoritma *FP-Growth* menggunakan *FP-Tree* (*Frequent pattern Growth*).

FP-Tree terdiri dari *root* yang diberi nilai *null*, *subtree* sebagai anak dari *root*, dan tabel *frequent header*. Node dalam *FP-Tree* mempunyai tiga informasi penting, yang terdiri dari *label item*, *support count,* dan *pointer*. *Label item*, menginformasikan jenis item dari node tersebut. *Support count*, menginformasikan jumlah lintasan transaksi yang melalui node tersebut. *Pointer,* merupakan penghubung node-node dengan label item yang sama antar lintasan. Metode *FP-Growth* dibagi menjadi tiga tahapan utama, antara lain :

1. Tahap Pembangkitan *Conditional Pattern Base*

   Tahap ini berisikan *prefix path* (lintasan *prefix*) dan *suffix pattern* (pola akhiran). Pembangkitan *Conditional Pattern Base* didapat dari *FP-Tree* yang telah dibangun sebelumnya.

2. Tahap Pembangkitan *Conditional FP-Tree*

   Pada tahap ini *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, item yang mempunyai jumlah *support count* lebih besar sama dengan *minimun support count* akan dibangkitkan dengan menggunakan *FP-Tree.*

3. Tahap Pencarian *frequent itemset*.

   Pada tahap ini jika kondisi *condotional FP-Tree* lintasan tunggal (*single path*), maka *frequent itemset* didapatkan dengan melakukan kombinasi item untuk setiap *conditional FP-Tree*. Sedangkan, jika bukan lintasan tunggal, maka pembangkitan *FP-Growth* secara rekursif.

Tahap-tahap tersebut merupakan tahapan untuk mendapatkan *frequent itemset* dalam algoritma *FP-Growth*.

```
Input : FP-Tree Tree
Output : Rt sekumpulan lengkap pola
frequent
Method : FP-growth (Tree, null)
Procedure : FP-growth (Tree, _)
{
01: if Tree mengandung single path P;
02: then untuk tiap kombinasi (dinotasikan _)
dari node-node dalam path do
03: bangkitkan pola _ _ dengan support dari
node-node dalam _;
04: else untuk tiap a1 dalam header dari Tree
do
{
05: bangkitkan pola
06: bangun _ = a1 _ dengan support = a1.
```

```
support
07: if Tree _ = _
08: then panggil FP-growth (Tree, _)
}
}
```

Gambar 2.2 *Pseudocode* algoritma *FP-Growth*

Sumber : [HAN-04]