BAB 2
TINJAUAN PUSTAKA

2.1 Profil Institusi

Makmur Jaya Kosmetik merupakan salah satu bidang usaha penjualan produk-produk kebutuhan kecantikan dan kebutuhan salon dari berbagai merk dalam negeri maupun merk dari luar negeri. Makmur Jaya Kosmetik berlokasi di Jln. Jend. A. Yani No.62 Garut

2.1.1 Sejarah Toko

Toko Makmur Jaya Kosmetik didirikan oleh ibu Hj. Ratna Rosita pada tahun 1986. Nama Makmur Jaya diambil dari nama anak beliau yang bernama Reka Makmur Jaya. Ide untuk mendirikan toko Makmur Jaya Kosmetik ini berawal dari banyaknya konsumen wanita yang sudah membutuhkan kosmetik dan pada saat itu toko kosmetik masih dibilang jarang. Toko Makmur Jaya Kosmetik awalnya adalah sebuah kios kecil biasa yang menjual merk dagang dalam negri. Seiring berkembangnya waktu, kebutuhan akan penunjang kecantikan khususnya bagi kaum wanita semakin bertambah. Pada saat ini Makmur Jaya Kosmetik sudah banyak menjual produk dalam dan luar negri seperti : Rista, Wardah, Ponds, LaTulipe, Tloac Paris, Tull Jye, Tje Fuk, jade, Bless dan lain-lain. Hingga akhirnya sampai sekarang Toko Makmur Jaya Kosmetik merupakan toko kosmetik terbesar dan terkemuka diwilayahnya.

2.1.2 Logo

Logo dari Toko terdapat gambar bunga dibagian kiri serta nama dibagian kanan. Lambang bunga sendiri diambil karena bunga identik dengan keindahan atau kecantikan. Untuk tulisan Makmur Jaya diambil dari nama toko dan untuk tulisan Kosmetik diambil dari bidang usaha yang dijalani. Warna merah muda untuk logo diambil karena warna merah muda identik dengan wanita. Logo dari toko dapat dilihat pada gambar 2.1.



Gambar 2. 1 *Logo Toko*

2.1.3 Visi dan Misi

Berikut adalah visi dan misi dari toko :

2.1.3.1 Visi

Visi dari toko adalah sebagai berikut:

1. Menjadi penyedia kosmetik yang aman, lengkap dan terpercaya, serta dapat memberikan pelayanan yang berkualitas

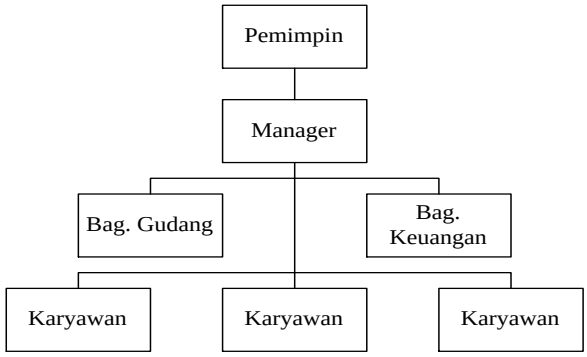
2.1.3.2 Misi

Misi dari toko adalah sebagai berikut:

1. Menyediakan produk kosmetik dari berbagai merk yang dibutuhkan oleh pelanggan
2. Menyediakan kosmetik yang aman yang memiliki label halal serta tercatat di Badan Pom
3. Memberikan pelayanan yang baik pada seluruh pelanggan

2.1.4 Struktur Organisasi

Struktur organisasi dari toko dapat dilihat dari gambar 2.2.



Gambar 2. 2 *Struktur Organisasi Toko*

Deskripsi Jabatan :

1. **Pemimpin**

   Pemimpin merupakan pemilik dari toko yang bertanggung jawab penuh terhadap segala sesuatu yang terjadi di toko .

2. **Manager**

   Manager merupakan seseorang yang membantu pemimpin dalam mengontrol atau mengawasi segala kegiatan di toko .

3. **Bag. Gudang**

   Bagian Gudang merupakan seseorang yang bertugas membantu pemilik dalam mengatur, mengelola, dan mencatat produk-produk yang ada digudang serta produk-produk yang akan dipesan untuk stok barang.

4. **Bag. Keuangan**

   Bagian Keuangan merupakan seseorang yang bertugas untuk membantu pemilik dalam mencatat produk yang terjual dan transaksi yang telah terjadi.

5. **Karyawan**

   Karyawan merupakan seseorang yang bertugas untuk melayani konsumen dalam melakukan pembelian dan transaksi pembayaran.

2.2 Landasan teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan melakukan penenlitian. Landasan teori yang diuraikan merupakan hasil studi literatur, buku-buku, maupun situs internet.

2.2.1 Data

Data adalah segala fakta, angka atau teks yang dapat diproses oleh komputer. Data dapat digunakan sebagai *input* dan menghasilkan sebuah informasi. Data adalah sesuatu yang belum memiliki arti dan masih membutuhkan suatu pengolahan. Dalam data terdapat himpunan data yang merupakan kumpulan dari objek dan atributnya. Atribut merupakan sifat atau karakteristik dari suatu objek yang biasanya dikenal sebagai *variabel, field,* karakteristik atau fitur.Salah satu himpunan data adalah *record* data, yaitu data yang terdiri

dari sekumpulan *record*, yang masing-masing terdiri dari satu set atribut yang tetap. Salah satu yang termasuk dalam tipe data *record* yaitu data transaksi. Data transaksi merupakan sebuah tipe khusus dari *record* data, dimana tiap *record* (transaksi) meliputi satu *set item* [3].

2.2.2 Basis Data

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain – lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol) [4].

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti[4] :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan *data* yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/ tabel/ arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.2.2.1 Data pada Basis Data dan Hubungannya

Ada 3 jenis data pada sistem *database*, yaitu [5] :

1. Data operasional dari suatu organisasi, berupa *data* yang tersimpan dalam basis data
2. Data masukan (*input data*), data dari luar sistem yang dimasukan melalui peralatan input (*keyboard*), yang dapat merubah data operasional.
3. Data kelauran (*output data*), berupa laporan melalui peralatan *output* sebagai hasil dari dalam sistem yang mengakses data operasional.

2.2.2.2 Keuntungan dan Kerugian Pemakaian Sistem Database

Keuntungan [5] :

1. Terpeliharanya keselarasan *data*

2. *Data* dapat dipakai secara bersama-sama.
3. Memudahkan penerapan standarisaasi dan batas-batas pengamanan.
4. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
5. Program /data *independent*.

Kerugian :

1. Mahal dalam implementasinya.
2. Rumit
3. Penanganan proses *recovery backup* sulit.
4. Kerusakan pada sistem basis data dapat mempengaruhi.

2.2.3 Database Mangement System

Database Management System atau disingkat DBMS adalah perangkat lunak (*software*) yang berfungsi untuk mengelola *database*. Mulai dari membuat *database* itu sendiri, sampai dengan prosesyang berlaku dalam *database* tersebut, baik berupa *entry*, *edit*, hapus, *query* terhadap data, membuat laporan dan lain sebagainya secara efektif dan efesien. Salah satu jenis DBMS yang sangat terkenal saat ini adalah *Relational* DBMS (RDBMS), RDBMS mempresentasikan data dalam bentuk tabel – tabel yang saling berhubugan. Sebuah tabel disususn dalam bentuk baris (*record*) dan kolom (*field*) [6].

2.2.4 Data Mining

Data mining adalah proses untuk mendapatkan informasi yang berguna dari gedung basis data yang besar. *Data mining* juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah *data mining* kadang disebut juga *knowledge discovery*. Salah satu teknik yang dibuat dalam *data mining* adalah bagaimana menelusuri data yang ada untuk membangun sebuah model, kemudian menggunakan model tersebut agar dapat mengenali pola data yang lain yang tidak berada dalam basis data yang tersimpan. Dalam *data mining*, pengelompokan data juga bisa dilakukan. Tujuannya adalah agar kita dapat megetahui pola universal data-data yang ada [7].

Secara umum, definisi *data mining* dapat diartikan sebagai berikut [8]:

1. Proses penemuan pola yang menarik dari data yang tersimpan dalam jumlah besar.
2. Ekstrasi dari suatu informasi yang berguna atau menarik (non-trivial, implisit, sebelumnya belum diketahui potensi kegunaannya) pola atau pengetahuan dari data yang di simpan dalam jumlah besar.
3. Eksplorasi dari analisa secara otomatis atau semiotomatis terhadap data data dalam jumlah besar untuk mencari pola dan aturan yang berarti.

2.2.4.1 Metode Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu [9] :

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat mengumpulkan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelesan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik daripada ke arah kategori. Model dibangun dengan record lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada dimasa mendatang. Contoh prediksi dalam bisnis dan penelitian adalah :

1. Prediksi harga beras dalam tiga bulan yang akan datang.
2. Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah dinaikan.

4. Klasifikasi

Dalam klasifikasi, terdapat terget variabel kategori. sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori yaitu: pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah :

1. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
2. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record,* pengamatan, atau memperhatikan dan membentuk *class* objek-objek yang memiliki kemiripan. *Kluster* adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan tidak memiliki kemiripan dengan *record-record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya *variabel* target dalam pengklusteran. pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari *variabel target*. Akan tetapi, *algoritma* pengklusteran mencoba untuk melakukan pembagian terhadap keselurahan data menjadi kelompok-kelompok yang memiliki kemiripan (*homogen*), yang mana kemiripan *record* dalam suatu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah :

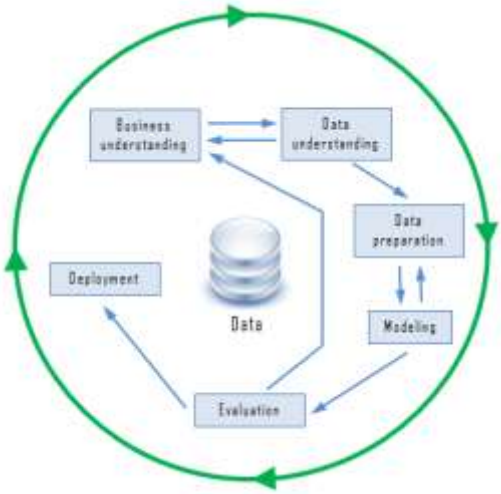
1. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk sebuah perusahaan yang tidak memiliki dana pemasaran yang besar.
2. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik maupun mencurigakan.
3. Asosiasi

   Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang pasar. Contoh asosiasi dalam bisnis dan penelitian adalah :

   1. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
   2. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

2.2.5 Cross-Industry Standard Process for Data Mining (CRISP-DM)

Cross-Indutry Standard Prosess for Data Mining (CRISP-DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industri seperti Daimler Chrysler, SPSS dan NCR. CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP-DM, sebuah proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan antarfase digambarkan dengan panah. Sebagai contoh, jika proses berada pada fase *modelling*. Berdasar pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuk perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation*. Enam fase yang ada dalam metode CRISP-DM tersebut dapat dilihat pada Gambar 2.2 :



Gambar 2. 3 CRISP – DM

Fase-fase dari CRISP-DM [11] :

1. *Business understanding*

   a. Penentuan tujuan objek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.

   b. Menilai situasi objek untuk mengetahui sumber daya yang tersedia.

   c. Menerjemahkan tujuan dan batasan menjadi formula dari permasalahan *data mining*.

2. *Data understanding*

   a. Mengumpulkan data.

   b. Menggunakan analisis penyelidikan data untuk mengenali lebih lanjut data dan pencarian pengetahuan awal.

   c. Mengevaluasi kualitas data.

   d. Jika diinginkan, pilih sebagian kecil grup data yang mungkin mengandung pola dari permasalahan.

3. *Data preparation*

a. Siapkan dari data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini merupakan pekerjaan berat yang perlu dilaksanakan secara intensif.

b. Pilih kasus dan variabel yang ingin dianalisis dan yang sesuai dengan analisis yang akan dilakukan.

c. Lakukan perubahan pada beberapa variabel jika dibutuhkan.

d. Sipakan data awal sehingga siap untuk perangkat pemodelan.

4. *Modelling*

a. Pilihan dan aplikasikan teknik pemodelan yang sesuai.

b. Membangun model yang digunakan.

c. Menilai model yang digunakan.

d. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu.

e. Aturan bisnis pembentukan paket

5. *Evaluation*

a. Mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan.

b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.

c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.

d. Mengambil keputusan berkaitan dengan penggunaan hasil dari *data mining*.

6. *Deployment*

a. Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.

b. Contoh sederhana penyebaran : pembuataan laporan.

c. Contoh kompleks penyebaran : penerapan proses *data mining* secara paralel pada departemen lain.

2.2.6 *Association Rule*

Aturan asosiasi *(Association rule)* adalah salah satu teknik tentang 'apa bersama apa'. Ini bisa berupa transaksi di supermarket, misalnya seseorang yang membeli susu bayi juga membeli sabun mandi. Di sini berarti susu bayi bersama sabun mandi. Karena awalnya berasal dari studi tentang *database* transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama produk apa, maka aturan asosiasi juga sering dinamakan *Market Basket* [10].

Untuk mencari *association rule* dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari *frequent itemset* terlebih dahulu. *Frequent itemset* adalah sekumpulan *item* yang sering muncul secara bersamaan. Setelah semua pola *frequent itemset* ditemukan, barulah mencari aturan asosiatif atau aturan keterkaitan yang memenuhi syarat yang telah ditentukan.

Jika diasumsikan bahwa barang yang dijual di swalayan adalah semesta, maka setiap barang akan memiliki *Boolean variable* yang akan menunjukan keberadaannya atau tidak barang tersebut dalam satu transaksi atau satu keranjang belanja. Pola *Boolean* yang didapat digunakan untuk menganalisa barang yang dibeli secara bersamaan. Pola tersebut dirumuskan dalam sebuah *association rule.* Sebagai contoh konsumen biasanya akan membeli kopi dan susu yang ditujukan sebagai berikut :

Kopi → susu [*support* = 2%, *confidence* = 60%]

Nilai *support* 2% menunjukan bahwa keseluruhan dari total transaksi konsumen membeli kopi dan susu secara bersamaan yaitu sebanyak 2%. Sedangkan *confidence* 60% yaitu menunjukan bila konsumen membeli kopi dan pasti membeli susu sebesar 60%.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, yaitu *support* dan *confidence. Support* (nilai penunjang) adalah presentase kombinasi *item* tersebut dalam *database*, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antar-*item* dalam aturan asosiasi.

Dalam menentukan suatu *association rule,* terdapat suatu ukuran kepercayaan yang didapat dari hasil pengelolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu:

1) *Support* : suatu ukuran yang menunjukan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *itemset* layak untuk dicari *confidence*-nya (misal, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi suatu *item* yang menunjukan bahwa *item* A dan *item* B dibeli bersamaan).
2) *Confidence* : suatu ukuran yang menunjukan hubungan antara 2 *item* secara *conditional* (misal, menghitung kemungkinan seberapa sering *item* B dibeli oleh pelanggan jika pelanggan tersebut membeli sebuah *item* A).

Kedua ukuran ini nantinya berguna dalam menentukan kekuatan suatu pola dengan membandingkan pola tersebut dengan nilai minimum kedua parameter tersebut yang ditentukan oleh pengguna. Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk *support* (*minimum support*) dan syarat minimum untuk *confidence* (*minimum confidence*).

Nilai *support* sebuah *item* diperoleh dengan rumus sebagai berikut[11] :

$$Support(\mathrm{A}) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi} \mathrm{x}100\% \ldots\ldots \text{ Persamaan (2.1)}$$

Sementara itu, nilai *support* dari 2 *item* diperoleh dari rumus berikut:

$$Support\ (\mathrm{A}, \mathrm{B}) = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi}} \mathrm{x}100\% \text{..Persamaan (2.2)}$$

Sedangkan nilai *confidence* dapat dicari setelah pola frekuensi munculnya sebuah *item* ditemukan. Rumus untuk menghitung *confidence* adalah sebagai berikut:

$$Confidence = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi yang mengandung A}} \text{x}100\% \text{.....Persamaan(2.3)}$$

2.2.7 Algoritma *FP-Growth*

Algoritma *fp-growth* merupakan salah satu alternatif algoritma yang cukup efektif untuk mencari himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data yang besar. Algoritma *fp-growth* merupakan algoritma *association rules* yang cukup sering dipakai. Algoritma *fp-growth* ini dikembangkan dari algoritma apriori. Algoritma apriori menghasilkan kombinasi yang sangat banyak sehingga sangat tidak efisien.

Algoritma *fp-growth* ini merupakan salah satu solusi dari algoritma apriori yang memakan waktu yang sangat lama karena harus melakukan *pattern matching* yang secara berulang-ulang. Sedangkan dalam proses algoritma *fp-growth* terdapat banyak kelebihan yang terbukti sangat efisien karena hanya dilakukan pemetaan data atau *scan* database sebanyak 2 kali untuk membangun struktur ”*tree*”. Maka dari itu, algoritma *fp-growth* dikenal juga dengan sebutan algoritma *FP-Tree* . Dengan menggunakan struktur *FP-Tree* , algoritma *fp-growth* dapat langsung mengekstrak *frequent itemset* dari susunan *FP-Tree* yang telah terbentuk. Metode *fp-grwth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai :

1) Tahap Pembangkitan *conditional pattern base.*

   Conditional Pattern Base merupakan sub *database* yang berisi *prefix path* dan *suffix pattern*. Pembangkitan *conditinal pattern base* didapatkan melalui *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, lalu setiap item yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support count* ξ akan dibangkitkan dengan *conditional FP-Tree* .

3) Tahap Pencarian *frequent itemset*

Apabila *conditional FP-Tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dnegan melakukan kombinasi item untuk setiap *conditional FP-Tree* . Jika bukan lintasan tunggal, maka dilakukan pembangkitan FP-Growth secara rekursif.

Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*,yang dapat dilihat pada algoritma berikut :

```
Input : FP-Tree Tree
Output : Rt Sekumpulan lengkap pola frequent
Method : FP-Ggrowth ( 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 daro 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, _ )
}
```

Contoh kasus penerapan Algoritma *FP-Growth*

-Misalkan didapat hasil dari tabel pembangkitan *conditional pattern base* sebagai berikut, diketahui sebelumnya nilai minimum support 5%.

a. Tahap Pembangkitan *Conditional Pattern Base*

Tabel 2. 1 Conditional Pattern Base

Item	*Conditional Pattern Base*
1	**2,3, 1:6**

b. Tahap Pembangkitan *Conditional FP-Tree*

Pada tahap ini, *support count* dari setiap item dijumlahkan, lalu setiapdan dibandingkan dengan *conditional fp - tree*

- Conditional fp – tree 1: { 2,4,3 }



Gambar 2. 4 Conditional FP – Tree 1

Karena *support count* 1 adalah 6, maka memenuhi *minimum support* yaitu 4.

Berikut adalah tahap-tahap pembangkitan *conditional FP-Tree* 1.

- Pada lintasan yang berakhiran {1} dengan *support count* 1 atau lebih yang dimilikinya akan dihilangkan dan *support count* yang melewati lintasan pertama atau lintasan lainnya yang berakhiran {1}, akn disamain sesuai dengan support count yang dimilikinya.



Gambar 2. 5 Conditional FP – Tree 1 Setelah Dibangkitkan

- Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count* akan selalu bertambah 1

Setelah proses pembangkitan, proses terakhir adalah mencari *frequent itemset* yang keluar. 2:6 dan 3:6 adalah *item* yang keluar



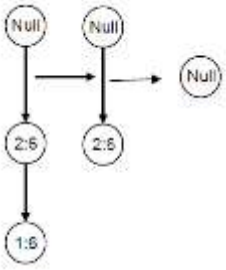
Gambar 2. 6 Conditional *FP-Tree* 1,2:6 Hasil Frequent Itemset



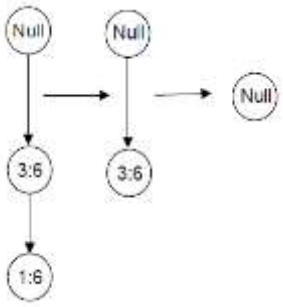
Gambar 2. 7 Conditional *FP-Tree* 1,3:6 Hasil Frequent Itemset

c. Tahap Pencarian *frequent itemset*

- Akumulasiakn count setiap item
- Buat *conditional FP-Tree*



Gambar 2. 8 Conditional *FP-Tree* Untuk Item Frequenset 1, 2:6



Gambar 2. 9 Conditional FP-Tree Untuk Item Frequenset 1, 3:6

2.2.8 *FP – Tree*

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. *FP-Tree* dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam *FP-Tree* . Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data *FP-Tree* semakin efektif. Kelebihan dari *FP-Tree* adalah hanya memerlukan dua kali pemindaian data transaksi yang terbukti sangat efisien. Adapun *FP-Tree* adalah sebuah pohon dengan definisi sebagai berikut :

1. *FP-Tree* dibentuk oleh sebuah akar yang diberi *label null*, sekumpulan pohon yang beranggotakan item-item tertentu, dan sebuah tabel *frequent header*.
2. Setiap simpul dalam *FP-Tree* mengandung tiga informasi penting, yaitu *label item*, menginformasikan jenis item yang dipresentasikan simpul

tersebut, *support count* merepresentasikan jumlah lintasan transaksi yang melalui simpul tersebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label item sama antar lintasan, ditandai dengan garis putus-putus.

2.2.9 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek serta aplikasinya. Berikut adalah beberapa model yang digunakan dalam perancangan Data Mining Pemaketan Produk untuk menggambarkan sistem dalam UML:

1. Diagram *Use Case*
2. Diagram Class
3. Diagram Aktivitas
4. Diagram Objek

2.2.9.1. Diagram *Use Case*

Diagram use case adalah model fungsional sebuah sistem yang menggunakan aktor dan *use case*. *Use case* adalah layanan (services) atau fungsi–fungsi yang disediakan oleh sistem untuk penggunanya.

Deskripsi Diagram Use Case:

1. Sebuah *use case* adalah dimana sistem digunakan untuk memenuhi satu atau lebih kebutuhan pemakai.
2. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design testing, dan dokumentasi.
3. *Use case* menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.
4. Use case menentukan nilai yang diberikan sistem kepada pemakainya.
5. *Use case* hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem.

6. *Use case* tidak untuk menentukan kebutuhan nonfungsional, misal: sasaran kerja, bahasa pemrograman.

2.2.9.2 Diagram *Class*

Diagram *class* adalah Diagram UML yang menggambarkan *class-class* dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Tahapan dari Diagram *class* adalah sebagai berikut:

1. Mengidentifikasi objek dan mendapatkan *class-class*nya.
2. Mengidentifikasi atribut *class-class*.
3. Mulai mengkonstruksikan kamus data.
4. Mengidentifikasi operasi pada *class-class*.
5. Mengidentifikasikan hubungan antar *class* dengan menggunakan *asosiasi*, *agregasi*, dan *inheritance* (pewarisan).

2.2.9.3 Diagram Aktifitas

Diagram aktivitas adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

2.2.9.4 Diagram Objek

Diagram objek adalah suatu Diagram yang berfungsi untuk mengatur atribut, objek dan hubungan antara contoh dalam Diagram. Diagram objek juga dapat menampilkan struktur model sistem dalam waktu tertentu. Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada Diagram objek harus dipastikan semua *class* yang sudah didefinisikan pada Diagram *class* harus dipakai objeknya, karena jika tidak, pendefinisian *class* itu tidak dapat dipertanggungjawabkan. Meskipun Diagram *class* diperlukan untuk mendokumentasikan struktur *class*, tetapi

Diagram objek, dapat berguna. Sebuah Diagram objek dasarnya instansiasi semua atau bagian dari Diagram *class*

2.2.10 MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *structured Query Language* (SQL). MySQL dalam operasi *client server* melibatkan *server daemon* MySQL di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client.* MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku bahwa MySQL yang mampu menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data [12].

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai bahasa yang standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan DBASE atau *Clipper* yang masih menggunakan perintah-perintah pemrograman.