BAB 2

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

PT.COPPAL UTAMA INDOMELT adalah Perusahaan yang bergerak dibidang pengecoran logam, tembaga, baja untuk membuat bagian atau rangka awal suatu produk yang dibutuhkan oleh perusahaan yang bergerak di pembuatan mobil, alat industri dan lainnya.

PT.Coppal Utama Indomelt didirikan tahun 1998 dan dibangun dari buah keterbatasan yang membuat semangat juang tumbuh semakin kuat, kami selalu berusaha menjaga kepuasan konsumen dengan terus mengembangkan penelitian dan teknologi di bidang pengecoran logam. Tujuan yang terus perusahaan ini kembangkan dari waktu ke waktu adalah membuat *casting* yang mempunyai kualitas baik dan harga yang bersaing. Perusahaan ini juga ingin menjadi sebuah perusahaan pengecoran logam yang disegani dan mempunyai pasar yang luas di Indonesia.

Perusahaan ini juga selalu memotivasi seluruh aspek yang ada di perusahaan agar mengutamakan kepuasan pelanggan. Komplain atau kritik dari pelanggan adalah salah satu prioritas perusahaan ini untuk terus mengadakan improvisasi kualitas pruduk dan manajemen.

2.1.1 Logo



Gambar 2 1 Logo Perusahaan

2.1.2 Visi dan Misi

2.1.2.1 Visi

“Smart Indonesia Cast metal manufacturing in Foundry Industry”

2.1.2.2 Misi

"*National solution for local component supply with competitive quality, cost and delivery in advance technology"*

2.2 Landasan Teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan melakukan penelitian. 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 computer. Data dapat digunakan sebagai input dan menghasilkan sebuah informasi. Data adalah seusuatu yang belum memiliki arti dan masih membutuhkan suatu pengolahan. Dalam data terdapat himpunan data yang merupakan kumpulan dari objek dan atributnya. Atributnya merupakan sifat atau 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 fata *record* yaitu data transaksi. Data transaksi merupakan sebuah tipe khusus dari *record* data, dimana tiap *record* (transaksi) meliputi satu *set item.*

2.2.2 Database

Database adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna.

2.2.2.1 Data pada Database dan Hubungannya

Ada 3 jenis data pada sistem *database,* yaitu:

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 keluaran (*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 :

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

Kerugian:

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

2.2.3 Data Mining

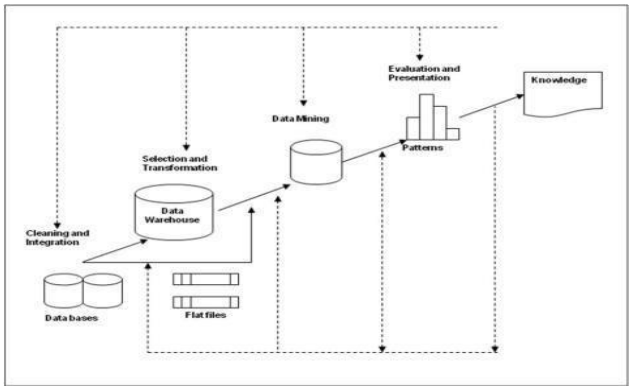
Data Mining adalah proses untuk mendapatan informasi yang berguna dari gedung basis data yang besar. *Data mining* juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah data mining kadang disebut juga *knowledge 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 mengetahui pola *universal* data-data yang ada.

Istilah *data mining* sering disebut dengan nama *knowledge discovery atau* penemuan pengetahuan karena tujuan utama dari *data mining* untuk mendapatkan pengetahuan yang masih tersembunyi di dalam bongkahan data. Secara umum, definisi *data mining* dapat diartikan sebagai berikut:

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 otmatis atau semiotomatis terhadap data data dalam jumlah besar untuk mencari pola dan aturan yang berarti.

2.2.3.1 Konsep *Data Mining*

Data mining sangat diperlukan terutama dalam mengelola data yang sangat besar untuk meudahkan aktifitas recording suatu transaksi dan untuk proses *data warehousing* agar dapat memberikan informasi yang akurat bagi pengguna *data mining*. Alasan utama *data mining* sangat dibutuhkan dalam industri informasi karena tersedianya data dalam jumlah yang besar dan semakin besarnya kebutuhan untuk mengubah data tersebut menjadi informasi dan pengetahuan yang berguna karena sesuai fokus bidang ilmu ini yaitu melakukan kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran atau berjumlah besar. Informasi iniliah yang nantinya sangat berguna untuk pengembangan. Berikut adalah langkah-langkah dalam *data mining* :



Gambar 2. 2 Konsep Data Mining

1. *Data cleaning* yaitu untuk menghilangkan *noise* data yang tidak konsisten.
2. *Data integration* yaitu menggabungkan beberapa *file* atau *database*.
3. *Data selection* yaitu data yang relevan dengan tugas analisis dikembalikan ke dalam *database* untuk proses *data mining*.
4. *Data transformation* yaitu data berubah atau bersatu menjadi bentuk yang tepat untuk menambang dengan ringkasan performa atau operasi agresi.
5. *Data mining* yaitu proses esensial dimana metode yang intelejen digunakan untuk mengekstrak pola data.
1. *Knowledge disccovery* yaitu proses esential dimana metode yang intelejen digunakan untuk mengekstrak pola data.
2. *Pattern evolution* yaitu untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik.
3. *Knowledge presentation* yaitu gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambah kepada *user*.

2.2.3.2 Tahapan *Data Mining*

Tahapan dalam melakukan *data mining* salah satunya adalah *preprocessing* data. Tahapan ini biasanya diperlukan karena data yang akan digunakan belum baik, yang disebabkan oleh beberapa faktor berikut ini :

1. *Incomplete* : tidak lengkapnya nilai suatu atribut, tidak lengkapnya atribut-atribut yang penting, atau hanya mempunyai data yang merupakan rekapitulasi
2. *Noisy* : mengandung *eror* atau merupakan *value* yang tidak wajar.
3. *Inconsisten* : mengandung nilai yang saling bertentangan.

Padahal data yang bisa diterima untuk bisa diproses menjadi informasi atau *knowledge* adalah data yang mempunyai kualitas diantaranya :

a. Akurat
b. Lengkap
c. Konsisten
d. Relevan

e. Bisa dipercaya
f. Mempunyai nilai tambah
g. Kemudahan untuk dimengerti

Jika data tidak dalam kualitas seperti yang telah diuraikan diatas,maka kualitas analisis data menajdi kurang sehingga hasilnya pun kurang bermakna. Hal tersebut harus dihindarkan karena hasil analisa yang salah dapat berujung pada solusi yang salah. Untuk itu, perlu dilakukan *preprocessing* data yang bertujuan agar membuat data menjadi lebih berkualitas. Adapun tahapan-tahapannya adalah sebagai berikut :

1. *Data cleaning* : mengisi/mengganti nilai-nilai yang hilang. Menghaluskan daya yang *noisy*, mengidentifikasi dan menghilangkan data yang tidak wajar, dan menyelesaikan masalah *inconsistensi* data.
2. *Data integration* : menggabungkan beberapa *database,* dan *file* menjadi satu sehingga didapatkan sumber data yang besar.
3. *Data transformation* : normalisasi dan agresi data.
4. *Data reduction* : mengurangi *volume* data namun tetap mempertahankan arti dalam hal hasil analisis data.
5. *Data discretization* : merupakan bagian dari data *reduction* dengan memperhitungkan data yang signifikan, khususnya pada data numerik.

2.2.3.3 Metode –Metode *Data Mining*

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

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.
3.

5. Pengklusteran

Pengklusteran merupakan pengelompokan record, pengamatan, atau memperhatikan dan membentuk kelas 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.

6. Asosiasi

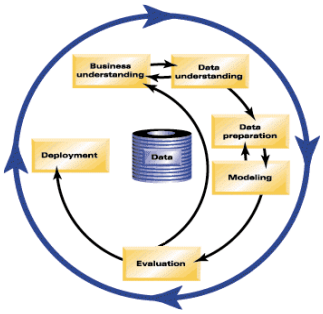
Tugas asosiasi dalam data mining adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang pasar. Contoh asosiasi dalam bisnis dan penelitian adalah :

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.

2.2.2.4 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 2.3 :



Gambar 2. 3 CRISP-DM

Fase-fase dari CRISP-DM [8] :

1. ***Business understanding***
   a. Penentuan tujuan objek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
   b. 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. Kalibrasi aturan model untuk mengoptimalkan hasil.

c. Perlu diperhatikan bahwa beberapa teknik mungkin untuk digunakan pada permasalahan *data mining* yang sama.

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.

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.

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

2.2.5 Association Rule

Association rule adalah salah satu teknik utama atau prosedur dalam *market basket analysis* untuk mencari hugungan antar-*item* dalam suatu *dataset* dan menampilkan bentuk *associaiton rule* [10]. *Association rule* (aturan asosiasi) akan menemukan pola tertentu untuk mengasosiasikan data yang satu dengan data yang lain.

Untuk mencari *association rule* dari suatu kumpulan data, tahap pertama yang ahrus dilakukan adalah mencari *frequent itemset* terlebih dahulu. *Frequent itemset* adalah sekumpulan *item* yang sering muncul secara bersamaan. Setelah semua pola *frequent item* ditemukan, barulah mencari aturan assosiatif 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* variabel yang akan menunjukkan keberadaannya atau tidak barang tersebut dalam suatu transaksi atau satu kerangajng 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 ditunjukkan sebagai berikut :

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

Association rule dperlukan suatu variabel ukuran yang ditentukan sendiri oleh *user* untuk menentukan batasan sejauh mana atau sebanyak apa *output* yang diinginkan *user*. *Support* dan *confidence* adalah sebuah ukuran kepercayaan dan kegunaan suatu pola yang telah ditemukan, Nilai *support* 2% menunjukkan bahwa keseluruhan dari total transaksi konsumen membeli kopi dan susu secara bersamaan yaitu sebanyak 2%. Sedangkan *confidence* 60%, yaitu menunjukkan bila konsumen membeli kopi dan pasti membeli susu sebesar 60%.

Penting tidaknya suatu aturan asosiasif dapat diektahui dengan dua parameter yaitu *support* dan iconfidence. *Support* (nilai pnunjang) adalah

persentase kombinasi *item* tersebut dalam *database*, sedangkan *confidencei* (nilai kepastian adalah kuatnya hubungan antar –*item* dalam aturan asosiasi.

Dalam menentukan suatu *associaiton rule*, terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didaapt dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu :

1. *Support* : suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item*/*itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *item*/*itemset* layak untuk dicari *confidence*-nya (misal, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi suatu item yang menunjukkan bahwa *item* A dan *item* B dibeli bersamaan).
2. *Confidence* : suatu ukuran yang menunjukkan 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. Bila suatu pola memenuhi kedua nilai minimum parameter yang sudah ditentukan sebelumnya, maka pola tersebut dapat disebut sebagai *interesting rule* atau *strong rule*. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap [12] :

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.

$$Support(A) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A}{Jumlah\ total\ transaksi} \text{ X 100 % ... Persamaan (2-1)}$$

Sementara itu, nilai *support* dari 2 *item* diperoleh dari rumus berikut $Support(A,B) = P(A \cap B)$…Persamaan (2-2) $Support(A,B) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ total\ transaksi}$ x100%.Persamaan(2-2)

2. Pembentukan aturan asosiasi

Setelah semua pola frekuensi tinggi ditemukan, kemudian mencari aturan asosiasi yang cukup kuat ketergantungan antar *item* .

Dalam *antecedent* (pendahulu) dan *consequent* (pengikut) serta memenuhi syarat *minimum* untuk *confidence* aturan asosiatif A→B.

Misalkan D adalah himpunan transaksi, dimana setiap transaksi T dalam D merepresentasikan himpunan *item* yang berada dalam I. I adalah himpunan *item* yang dijual. Misalkan kita memilih himpunan *item* A dan himpunan *item* lain B, kemudian aturan asosiasi akan berbentuk :

Jika A, maka B (A→B)

Dimana *antecedent* A dan *consequent* B merupakan subset dari I, dan A dan B dimana aturan :

Jika A, maka B

Tidak berarti

Jika B, maka A

Sebuah itemset adalah himpunan *item-item* yang ada dalam I, dan i *itemset*. Frekuensi itemset merupakan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai *minimum* yang telah ditentukan.

Nilai *confidence* dari aturan A→ B diperoleh dari rumus berikut.

$$Confidence = P(B \mid A) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ transaksi\ yang\ mengandung\ A}$$ Persamaan (2-3)

2.2.6 Algoritma FP-Growth

Algoritma yang sama dengan Apriori, FP-Growth mulai dengan menghitung item tunggal sesuai dengan jumlah kemunculan item yang ada didalam *dataset*. Setelah proses penghitungan selesai maka akan dibuat struktur pohon pada tahap kedua. Pohon yang dibuat mulanya kosong yang nanti akan diisi dengan hasil dari dataset yang telah didapat sebelumnya. Kunci untuk mendapatkan struktur pohon yang bisa didapatkan dengan proses lebih cepat untuk mencari item set yang besar menjadi sedikit dengan di urutkan secara

descending dari frekuensi yang ada *dataset* tersebut. Masing-masing item yang tidak mencapai kebutuhan minimum dari *threshold* tidak dimasukkan kedalam pohon, tapi dikeluarkan secara efektif dari *dataset* [8].

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:

a. FP-Tree dibentuk oleh sebuah akar yang diberi label *null*, sekumpulan berupa pohon yang beranggotakan item-item tertentu, dan sebuah tabel *frequentheader*.
b. Setiap simpul dalam FP- Tree mengandung tiga informasi penting, yaitu label item, menginformasikan jenis item yang direpresentasikan simpul tersebut, *support count*, merepresentasikan jumlah lintasan transaksi yang melalui simpul tesebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label item sama antar-lintasan, ditandai dengan garis panah putus-putus.

2.2.4.1 Langkah-Langkah Proses Perhitungan Association Rule Dengan Algoritma FP-Growth

Proses perhitungan *association rule* terdiri dari beberapa tahap adalah sebagai berikut [9] :

1. Membuat *Header Item*

   Header dalam hal ini selain sebagai *header* suatu *item* ke FP-Tree juga sebagai jenis *item* dasar yang memenuhi minimum *support*. Setelah mendapatkan *item* dan nilai *support*-nya, maka *item* yang tidak *frequent* dibuang dan *item* diurutkan berdasarkan nilai *support*-nya. Header

untuk item, disiapkan pada suatu *array* tertentu dan ditambahkan ketika membuat FP-Tree.

2. Membuat FP-Tree

   FP-Tree dibangun dengan mencari *item* sesuai urutan pada *item* yang *frequent*. Data transaksi tidak perlu diurutkan, dan untuk tiap *item* yang ditemukan bisa langsung dimasukkan ke dalam FP-Tree. Sesudah membuat *root*, tiap *item* yang ditemukan dimasukkan berdasarkan *path* pada FP-Tree. Jika *item* yang ditemukan sudah ada, maka nilai *supportitem* tersebut yang ditambahkan. Namun jika *path* belum ada, maka dibuat *node* baru untuk melengkapi *path* baru pada FP-Tree tersebut. Hal ini dilakukan selama item pada transaksi masih ada yang *qualified*, artinya memenuhi nilai minimum *support*. Jadi, *item-item* yang ditemukan dalam transaksi akan berurutan memanjang ke bawah. Dalam struktur FP-Tree, diterapkan alur *path* dari *child* hingga ke *root*. Jadi, suatu *path* utuh dalam FP-Tree adalah dari *child* terbawah hingga ke *root*. Tiap *node* pada FP-Tree memiliki *pointer* ke *parent*, sehingga pencarian harus dimulai dari bawah.

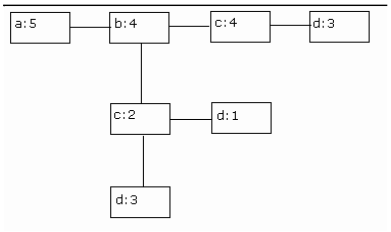
3. *Pattern Extraction*

   Pattern extraction dilakukan berdasarkan keterlibatan *item* pada suatu *path*. Di setiap *path*, diperiksa semua kombinasi yang mungkin dimana *item* tersebut terlibat. Di iterasi berikutnya dilakukan dengan melibatkan *item* berikutnya, tanpa melibatkan *item* sebelumnya, sehingga *pattern* yang sama tidak akan ditemukan dua kali pada *path* yang sama. Bila *item* pertama suatu hasil kombinasi bukan *item* terakhir (sebelum *root*), maka kombinasi *itemset* tersebut masih bisa dikembangkan lagi.

4. Memasukkan setiap pattern yang ditemukan ke dalam *PatternTree*

   Setelah mengolah FP-Tree menjadi *pattern-pattern*, diperlukan proses akumulasi *pattern-pattern* yang ditemukan mengingat *pattern* yang sama dapat ditemukan pada path yang berbeda. Untuk itu digunakan struktur data *Pattern Tree* (lihat Gambar 2.4). Setiap node

di *Pattern Tree* merepresentasikan dan menyimpan frekuensi suatu *pattern*. *Pattern Tree* terdiri atas *Pattern TreeNode* yang menyimpan nilai item, nilai *support* dan dilengkapi dengan dua *pointer* yaitu untuk horisontal dan vertikal.



Gambar 2. 1 Pattern Tree

Misalnya pada *noc* *pattern* a-c-d bernilai *support* 1. Kemudian bila ada pattern a-c-d lagi bernilai *support* n yang ditemukan dari FP-Tree maka nilai *support* 1 tersebut menjadi n+1. Contoh hasil lengkap dari *PatternTree* tersebut:

1. a:5 menggambarkan bahwa ada pattern a sebanyak 5
2. b:4 menggambarkan bahwa ada pattern a-b sebanyak 4
3. c:4 menggambarkan bahwa ada pattern a-b-c sebanyak 4
4. d:3 menggambarkan bahwa ada pattern a-b-c-d sebanyak 3
5. c:2 menggambarkan bahwa ada pattern a-c sebanyak 2
6. d:1 menggambarkan bahwa ada pattern a-c-d sebanyak 1
7. d:3 menggambarkan bahwa ada pattern a-d sebanyak 3

5. Mengurutkan dan Menyeleksi *Pattern*

   Pattern yang tidak memenuhi minimum *support*, dihapus dari daftar *pattern*. *Pattern-pattern* yang tersisa kemudian diurutkan untuk memudahkan pembuatan *rules*.

2.2.7 *Unified Modelling 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 di Minimarket Warga Tunggal untuk menggambarkan sistem dalam UML:

1. Diagram *Use Case*
2. Diagram Kelas
3. Diagram Aktivitas
4. Diagram Sequence

2.2.7.1 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.

2.2.7.2 Diagram Kelas

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

1. Mengidentifikasi objek dan mendapatkan kelas-kelasnya.
2. Mengidentifikasi atribut kelas-kelas.

3. Mulai mengkonstruksikan kamus data.
4. Mengidentifikasi operasi pada kelas-kelas.
5. Mengidentifikasikan hubungan antar kelas dengan menggunakan *asosiasi*, *agregasi*, dan *inheritance* (pewarisan).

2.2.7.3 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.

2.2.7.4 Diagram Sequence

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak. Penomoran pesan berdasarkan urutan iteraksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.