BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

Berikut adalah profil dari tempat penelitian Tugas Akhir.

2.1.1 Sejarah Distro Fresh wearhouse

Distro Fresh wearhouse merupakan salah satu distro yang menyediakan berbagai produk yang memenuhi kebutuhan trend anak muda di zaman yang serba modern ini. Distro ini didirikan di kawasan Jl. Raya Batujajar Timur No.148 Kab., Bandung Barat oleh Gugi Basmalah pada tahun 2008. Selain menjual kebutuhan trend anak muda dalam jumlah satuan, di distro ini juga menjual produk dalam bentuk paket yang pembaruannya dilakukan dalam beberapa event yang dijual lebih murah. Distro ini menerapkan konsep pembelian langsung yang dapat memudahkan pembeli dapat melihat dan memilih langsung produk yang akan dibeli sesuai dengan keinginan dan kebutuhannya. Hal inilah yang menjadikan distro ini sangat diminati oleh para konsumen.

Seiring dengan minat konsumen terhadap produk di distro Fresh Wearhouse ini setiap tahunnya maka pihak distro menambah beberapa produk yang sebelumnya tidak ada di distro. Dahulu distro ini hanya mempunyai beberapa jenis produk saja namun kini distro Fresh Wearhouse sudah memiliki beberapa jenis produk yang dapat konsumen beli. Pihak distro berencana untuk membuka cabang baru di daerah lainnya.

2.1.2 Logo

Berikut ini adalah logo. Bisa dilihat pada gambar 2.1:

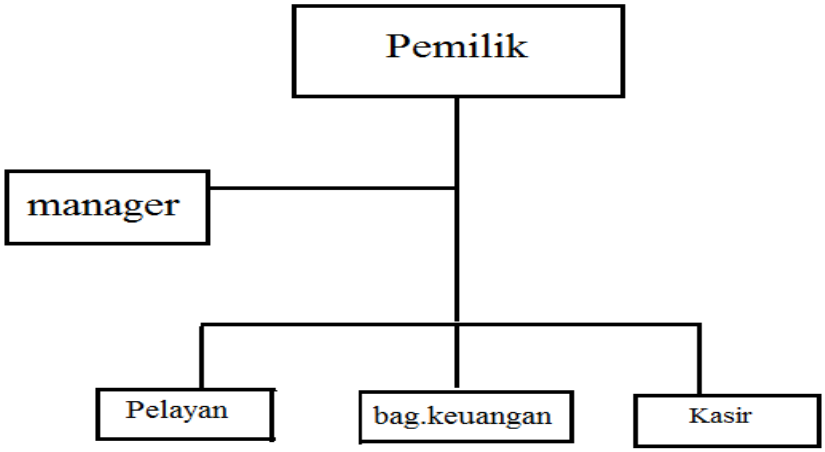


Gambar 2. 1 Logo Distro

Logo Fresh Wearhouse ini dibuat oleh pemilik distro dengan lambang fresh dengan maksud agar semua produk yang dihasilkan akan selalu fresh dimata konsumen seperti namanya.

2.1.3 Struktur Organisasi dan Job Description

Berikut ini adalah Struktur organisasi Distro Fresh Wearhouse Batujajar



Gambar 2. 2 Struktur Organisasi

Deskripsi Tugas :

1. Pemilik bertanggung jawab penuh terhadap distro, berikut ini adalah tugas dan tanggung jawabnya :

   a. Membuat perencanaan dan kebijakan yang menyangkut aktifitas distro.

b. Melakukan kontrol secara keseluruhan terhadap operasional distro.

c. Memegang kendali atas keputusan penting yang berhubungan dengan keuangan.

2. Manager berfungsi untuk membantu pemilik dalam menjalankan operasional distro berikut ini adalah tugas dan tanggung jawabnya :

a. Mengatur setiap bagian yang ada di distro agar menjalankan tugasnya dengan baik

b. Membantu pemilik dalam mengawasi operasional distro.

3. Bagian pelayan berfungsi untuk melayani setiap pelanggan yang datang atau memesan via online dan tanggung jawabnya adalah sebagai berikut :

a. Melayani semua pesanan pelanggan

b. Membantu dalam melayani pesanan pesanan pelanggan

4. Bagian Keuangan bertugas mengawasi transaksi pelanggan tugas dan tanggung jawabnya adalah sebagai berikut :

a. Menghitung pemasukan dan pengeluaran setiap bulannya

b. Bertanggung jawab terhadap hal-hal yang menyangkut keuangan di distro

5. Kasir berfungsi melayani transaksi pembelian tugas dan tanggung jawabnya adalah sebagai berikut :

a. Melayani transaksi pembayaran langsung dengan pelanggan

b. Menghitung dan mencatat pembelian setiap harinya.

2.1.4 Visi dan Misi

Visi dari distro Fresh Wearhouse ini adalah mendirikan distro dengan konsep modern dan menjadi distro yang mendistributor lokal dan berkedudukan berkepanjangan. Sedangkan misi dari distro Fresh Wearhouse ini adalah:

1. Melakukan pelayanan terbaik untuk pelanggan dan menjual kualitas produk yang akan dijual
2. Selalu melakukan inovasi terhadap produk
3. Mengembangkan usaha distro di wilayah Kab.Bandung Barat.
4. Meningkatkan Produktifitas dar Sumber Daya Manusianya agar lebih baik lagi

2.2 Landasan Teori

Dalam bab ini akan dijelaskan mengenai definisi dan teori-teori yang berkaitan dengan permasalahan yang ada. Landasan teori yang diuraikan merupakan hasil studi literatur, buku-buku, maupun internet.

2.2.1 Data

Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya[5]. *Data* dapat juga digunakan sebagai input dan menghasilkan sebuah informasi. *Data* juga merupakan 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 *variable, field,* karakteristik atau fitur. Salah satu himpunan *data* adalah *record data* , yaitu data yang terdiri dari sekumpulan *record* yang masing-masing terdiri dari suatu 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* [6].

2.2.2 Basis Data

Basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau *data*. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data,pengguna dapat menyimpan *data* secara terorganisasi. Setelah *data* disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi . cara *data* disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. *Data* pun harus mudah ditambahkan kedalam basis data, dimodifikasi, dan dihapus Menurut Fathansyah [5], basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

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 [9] :

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 opersional.
3. Data keluaran (*output data),* berupa laporan melalui peralatan *output* sebagai hasil dari sistem yang mengkses data operasional.

2.2.2.2 Keuntungan dan Kerugian Pemakaian Sistem Database

Keuntungan [10] :

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

Kerugian [10] :

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

2.2.3 Database Management system

Kumpulan atau gabungan *database* dengan perangkat lunak aplikasi yang berbasis *database* tersebut dianamakan *Database Management System (DBMS). DBMS* merupakan koleksi terpadu dari *database* dan program – program komputer (utilitas) yang digunakan untuk mengakses dan memelihara *database.* Progran-program tersebut menyediakan berbagai fasilitas operasi untuk memasukan, melacak, dan memodifikasi *data* kedalam *databse,* mendefinisikan *data* baru, serta mengolah data menjadi informasi yang dibutuhkan (*DBMS= Database* + program Utilitas) [10].

2.2.4 Data Mining

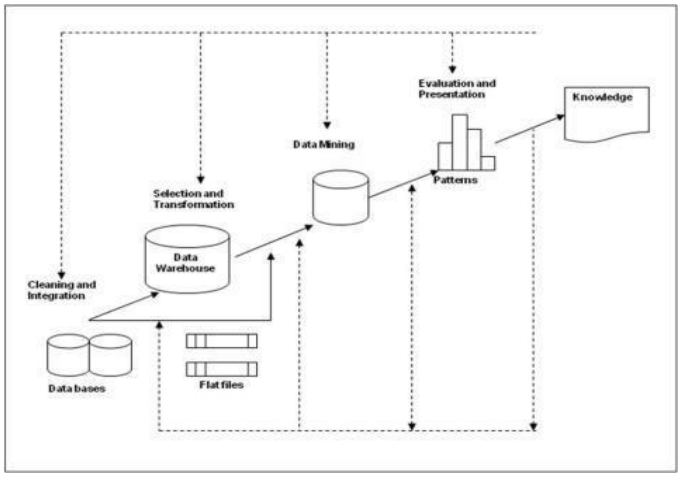
Data mining, sering juga disebut *knowledge discovery in database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian, dan historis untuk menemukan keteraturan, pola atau hubungan dalam *set data* berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan, sehingga istilah *pattern recognition* sudah tidak digunakan lagi [12].

Secara umum, definisi *data mining* dapat diartikan sebagai berikut [12]:

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 Konsep Data Mining

Data Mining sangat diperlukan terutama dalam mengelola data yang sangat besar untuk memudahkan aktifitas *recording* suatu transaksi dan untuk proses *data warehousing* agar dapat memberikan informasi yang akurat bagi pengguna *data mining.* Alasan utama *data mining* ssangat dibutuhkan dalamindustri 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 ekstrasi atau menambah pengetahuan dari data yang berukuran ataau berjumlah besar. Informasi inilah yang nantinya sangat berguna untuk pengembangan. Berikut adalah langkah-langkah dalam *data mining*[12]:



Gambar 2. 3 Konsep *Data Mining*

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

S2.2.4.2 Metode- Metode Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu [12]:

1. Deskripsi

   Terkadang peneliti dan analis secara sederhana ingin mecoba 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 pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola kecenderungan.

2. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik daripada kearah kategori. Model dibangun dengan *record* lengkap yang menyediakan nilai dari variable 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 akhir 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.

4. Klasifikasi

   Dalam klasifikasi, terdapat target 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 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 dengan 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 keseluruhan 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 akutansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik maupun mencurigakan.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam suatu 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 dibeli bersamaan.

2.2.5 CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan suatu standar yang telah dikembangkan pada tahun 1996 yang ditujukan untuk melakukan proses analisis dari suatu industri sebagai strategi pemecahan masalah dari bisnis atau unit penelitian. Untuk data yang dapat diproses dengan CRISP-DM ini, tidak ada ketentuan atau karakteristik tertentu, karena data tersebut akan diproses kembali pad fase-fse didalamnya. Terdapat enam fase dalam CRISP-DM ini yakni [6] :

1. Business Understanding

   Tahap awal ini berfokus pada penentuan tujuan bisnis, menilai situasi dan penentuan sasaran *data mining*
2. *Data understanding*

   Tahap pemahaman data dimualai dengan pengumpulan data awal, menjelaskan data yang memungkinkan kita untuk untuk lebih mengenal data
3. *Data preparation*

   Tahap persiapan data mencakup semua kegiatan yang diperlukan untuk membangun *data set* akhir dari data mentah awal. Tugas perssiapan data adalah pemilihan atribut data, *cleaning data*, pembangunan data, format data dan dekripsi data set.

4. *Modeling*

   Pada tahap ini akan dilakukan pemilihan teknik pemodelan, menghasilkan desain uji, pembangunan model, serta menilai model yang telah dibuat.

5. *Evaluation*

   Pada tahap ini kita harus mengevaluasi hasil, review proses, dan sebelum sampai akhir pembuatan model penting untuk benar-benar mengevaluasi dan meninjau kembali langkah dan pembuatannya, untukmemastikan bahwa model benar-benar mencapai tujuan bisnis. Tujuan utama adalah untuk menentukan apakah ada beberapa masalah bisnis yang belum cukup dipertimbangkan. Pada akhir dari tahap ini, keputusan tentang penggunaan hasil *data mining* harus tercapai.

6. *Deployment*

   Pada tahap ini, informasi yang diperoleh akan dipresentasikan dalam bentuk khusus sehingga dapat digunakan oleh pengguna. Tahap *deployment* dapat berupa pembangunan atau penginstalan software, pemantauan dan pemeliharaan, menghasilkan suatu laporan. Dalam beberapa kasus, tahap ini melibatkan konsumen, disamping analisis data karena sangat penting bagi konsumen untuk memahami tindakan apa yang harus dilakukan untuk menggunakan model yang telah dibuat.

2.2.6 *Association Rule*

Association Rule atau aturan asosiasi adalah teknik *data mining* untuk menemukan aturan asosiatif atau pola kombinasi dari suatu *item.* Bila kita mengambil contoh aturan asosiatif dalam suatu pembelian transaksi pembelian di minimarket adalah kita dapat mengetahui berapa besar kemungkinan seorang konsumen membeli suatu item bersamaan dengan item lainnya(membeli roti bersama selai) karena awalnya berasal dari studi tentang *database* transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli secara bersama apa, maka *association rule* sering disebut juga *market analysis bisnis*.

Association rule adalah bentuk jika "kejadian sebelumnya" kemudian "konsekuensinya" (*if antecedent, then consequent),* yang diikuti dengan perhitungan aturan *support* dan *confidence.* Bentuk umum dari *association rule* adalah *Antecedent -> Consequent.* Bila kita ambil contoh dalma sebuah transaksi pembelian barang di sebuah minimarket didapat bentuk *association rule* roti ->selai. Yang artinya bahwa pelanggan yang membeli roti ada kemungkinan pelanggan tersebut juga membeli selai, dimana tidak ada batasan dalam jumlah item-item pada bagian *antecedent* atau *consequent* dalam sebuah *rule.* Dalam menentukan suatu *association rule,* terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapat 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. *Support* merupakan matrik pertama yang ditetapkan dalam analisis keranjang pasar, yang merupakan probabilitas dari asosiasi (probabilitas dari dua item yang diberi bersama- sama). *Support* dihasilkan dari beberapa kali jumlah *item* A dan B terjadi bersamaan dalam transaksi yang sama dibagi dengan jumlah total dari transaksi tersebut. *Support* dapat dirumuskan sebagai berikut [15] :

$$Support = P(A \cap B)$$
$$= \frac{\text{Jumlah transaksi yang memuat A dan B}}{\text{Total jumlah transaksi}}$$

2. *Confidence* : *Confidence* dihasilkan dari seberapa kuat hubungan produk yang sudah dibeli. *Confidence* dapat dirumuskan sebagai berikut [15]:

$$Confidence = P(B / A)$$
$$= \frac{Support\ (A \cap B)}{P\ (A)}$$

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 yaitu [15] :

1. Analisis pola frekuensi tinggi

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

$$Support(A) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A}{Jumlah\ total\ transaksi} \quad \text{X 100 \%}$$

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(2-1)

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 assosiatif

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 rumus berikut :

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

Persamaan (2-2)

2.2.7 Algoritma FP-Growth

Algoritma yang sama dengan Apriori, FP-Growth mulai dengan menghitung *item* tunggal sesuai dengan jumlah kemunculan *item* yang ada dalam *dataset.* Setelah proses perhitungan 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 denagn proses lebih cepat untuk mencari *itemset* yang besar menjadi sedikit dengan diurutkan secara *descending* dari *frekuensi* yang ada *dataset* tersebut. Masing-masing *item* yang tidak mencapai kebutuhan minimum dari *threshold* tidak dimasukkan kedalam pohon, tapi tidak dikeluarkan secara efektif dari *dataset* [4].

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. FP-Tree dibangun dengan meletakan setiap data transaksi kedalam 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 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 *frequent header*
b. Setiap simpul dalam FP-Tree mengandung tiga informasi penting, yaitu label item, menginformasikan jenis item yang direpesentasikan simpul tersebut, *support count*, merepresentasikan jumlah lintasan transaksi melalui simpul tersebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label atau 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

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 *support item* 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*.

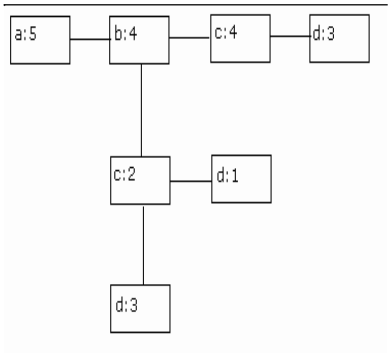
Tiap *node* pada FP-Tree memiliki *pointer* ke *parent*, sehingga pencarian harus dimulai dari bawah.

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 melakukan *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 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 2.5). 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 horizontal dan vertikal.



Gambar 2.4 Pattern Tree

Misalnya pada *node* d:1 di atas, berarti terdapat *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*.

2.2.8 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek serta aplikasinya.

2.2.8.1 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 dari *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.

2.2.8.2 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).

2.2.8.3 Diagram Aktifitas

Diagram aktifitas 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.

2.2.8.4 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.