BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

2.1.1 Sejarah Minimarket Warga Tunggal

Minimarket Warga Tunggal merupakan salah satu Minimarket yang menyediakan berbagai produk untuk memenuhi kebutuhan sehari - hari. Minimarket Warga Tunggal ini didirikan di Jalan Karees Timur Nomor 22 Bandung oleh Sri Andayani. Selain menjual produk – produk untuk kebutuhan sehari – hari dalam bentuk satuan Minimarket Warga Tunggal ini juga menjual produk untuk kebutuhan sehari – hari dalam bentuk Paket Produk dengan harga jual lebih murah daripada membeli produk dalam bentuk satuan. Minimarket Warga Tunggal ini menerapkan konsep swalayan dalam proses penjualannya, sehingga memberikan kemudahan bagi para konsumen untuk melihat dan memilih secara langsung produk yang akan dibeli sesuai dengan keinginan dan kebutuhannya. Selain itu, konsumen juga dapat membeli produk dalam jumlah yang banyak atau grosir dengan harga yang lebih murah. Hal inilah yang menjadikan Minimarket Warga Tunggal ini sangat diminati oleh para konsumen.

Seiring dengan minat konsumen terhadap Minimarket Warga Tunggal yang semakin meningkat setiap tahunnya, pihak Minimarket berencana untuk membuka cabang baru di daerah lain.

2.1.2 Logo

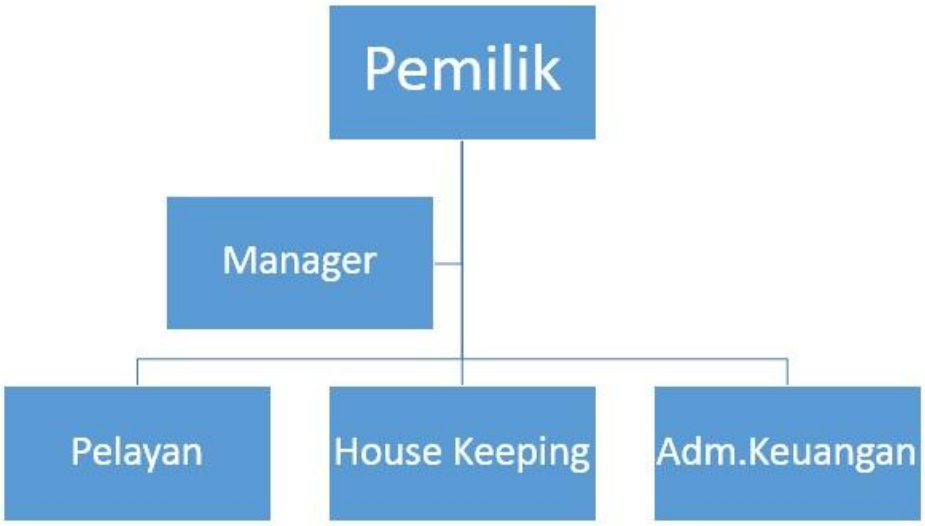
Berikut ini adalah logo dari Minimarket Warga Tunggal yang dapat dilihat pada Gambar 2.1 :



Gambar 2. 1 Logo Minimarket Warga Tunggal

2.1.3 Struktur Organisasi

Berikut ini adalah Struktur Organisasi Minimarket Warga Tunggal yang dapat dilihat pada Gambar 2.2 :



Gambar 2. 2 Struktur organisasi Minimarket Warga Tunggal

Job description :

1. Pemilik bertanggung jawab penuh terhadap Minimarket, tugas dan tanggung jawabnya sebagai berikut :
    a. Membuat perencanaan, strategi dan kebijakan yang menyangkut operasi Minimarket.
    b. Menyusun anggaran kebutuhan persediaan produk.
    c. Melakukan kontrol secara keseluruhan atau operasi Minimarket.
    d. Memegang kendali atas keputusan penting yang bersifat umum berkaitan dengan keuangan.
2. Manager berfungsi untuk membantu pemilik, tugas dan tanggung jawabnya adalah sebagai berikut :
    a. Membantu pemilik dalam mengawasi Minimarket.
    b. Mengatur setiap bagian yang ada di Minimarket agar menjalankan tugasnya dengan baik.
3. Bagian pelayanan berfungsi untuk melayani tiap pelanggan, tugas dan tanggung jawab sebagai berikut :
    a. Membantu dalam melayani pesanan-pesanan pelanggan.
    b. Melayani semua pesanan pelanggan.
4. Bagian Adm.Keuangan bertugas terhadap transaksi pelanggan, tugas dan tanggung jawab sebagai berikut :
    a. Bertanggung jawab terhadap hal-hal yang menyangkut keuangan.
    b. Menghitung pemasukan dan pengeluaran setiap bulannya.
    c. Melayani pelanggan dalam pembayaran transaksi.
5. Bagian *house keeping* bertugas terhadap kebersihan dan kenyamanan pelanggan, tugas dan tanggung jawab sebagai berikut :
    a. Bertanggung jawab terhadap kebersihan, baik di luar ataupun di dalam lingkungan Minimarket.
    b. Membereskan dan membersihkan peralatan yang telah digunakan.

2.1.4 Visi dan Misi

Visi dari Minimarket Warga Tunggal adalah Mendirikan Minimarket dengan konsep Modern dan menjadi Minimarket distributor lokal yang berkedudukan dan berkepanjangan. Sedangkan Misi dari Minimarket Warga Tunggal adalah sebagai berikut :

1. Mengutamakan pelayanan terbaik untuk pelanggan serta kualitas makanan yang akan dijual.
2. Melakukan inovasi terhadap produk dan sistem penjualan secara berkala.
3. Mengembangkan usaha Minimarket di luar wilayah Bandung.
4. Meningkatkan produktifitas Minimarket dari Sumber Daya. Manusianya agar Minimarket terus berkembang ke arah yang lebih baik.

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 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 [10]. *Data* dapat 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 *variabel, field,* karakteristik atau fitur.Salah satu himpunan *data* adalah *recorddata*, yaitu *data* yang terdiri dari sekumpulan *record*, yang

masing-masing terdiri dari satu set atribut yang tetap. Salah satu yang termasuk dalam tipe *datarecord* yaitu *data* transaksi. *Data* transaksi merupakan sebuah tipe khusus dari *record data*, dimana tiap *record* (transaksi) meliputi satu *setitem* [11].

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 ke dalam basis data, dimodifikasi, dan dihapus [12].

Menurut Fathansyah [14], 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 [13] :

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 Pemakain Sistem Database

Keuntungan [13] :

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 [13]:

1. Mahal dalam implementasinya.
2. Rumit
3. Penanganan proses *recoverybackup* 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 dinamakan *Database Management System (DBMS).DBMS* merupakan koleksi terpadu dari *database* dan program–program komputer (utilitas) yang digunakan untuk mengakses dan memelihara *database*. Program-program tersebut menyediakan berbagai fasilitas operasi untuk memasukan, melacak, dan memodifikasi data kedalam *database*, mendefinisikan *data* baru, serta mengolah data menjadi informasi yang dibutuhkan (*DBMS* = *Database* + Program Utilitas) [14].

2.2.4 *Data Mining*

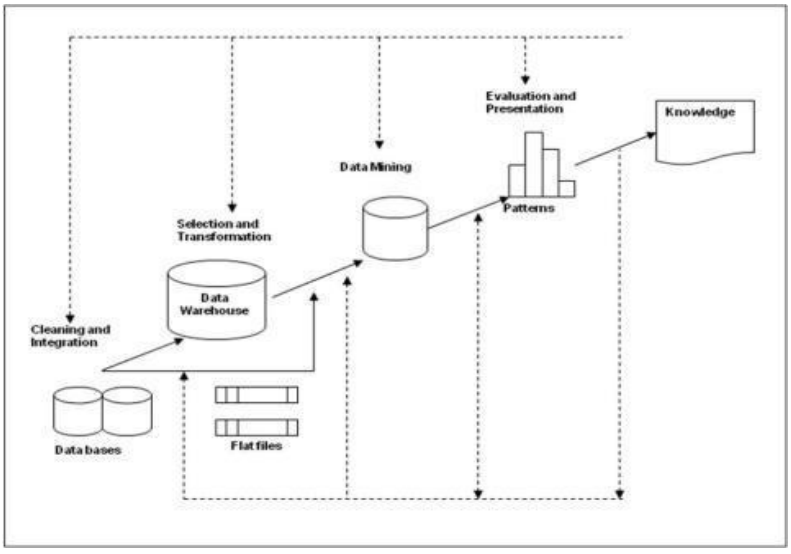
Data mining, sering juga disebut *knowledge discovery in database (KDD)*, adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set *data* berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan. Sehingga istilah *pattern recognition* sudah tidak digunakan lagi [15].

Secara umum, definisi *data mining* dapat diartikan sebagai berikut [16]:

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 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 [15] :



Gambar 2. 3 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*.
6. *Knowledge disccovery* yaitu proses esential dimana metode yang intelejen digunakan untuk mengekstrak pola *data*.
7. *Pattern evolution* yaitu untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik.
8. *Knowledge presentation* yaitu gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambah kepada *user*.

2.2.4.1 Metode –Metode Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu [17] :

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari carauntuk 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 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.

2.2.5 *Association Rule*

Association Rule atau Aturan Asosiasi adalah teknik *Data Mining*untukmenemukan aturan asosiatif atau pola kombinasi dari suatu *item*. Bila kita mengambil contoh aturan asosiatif dalam suatu transaksi pembelian barang di suatu minimarket adalah kita dapat mengetahui berapa besar kemungkinan seorang konsumen membeli suatu item bersamaan dengan item lainnya (membeli roti bersama dengan selai). Karena awalnya berasal dari studi tentang *database* transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama apa, maka *association rule* sering juga dinamakan *market basket analysis* [15].

Association Rule adalah bentuk jika "kejadian sebelumnya" kemudian "konsekuensinya" (*If antecedent, then consequent*), yang diikuti denganperhitungan aturan *support* dan *confidence*. Bentuk umum dari *association rule*adalah *Antecedent -> Consequent.*Bila kita ambil contoh dalam 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 akan membeli selai, dimana tidak ada batasan dalam jumlah item-item pada bagian antecedent ataupun *consequent* dalam sebuah *rule*. 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. *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 berapa kali jumlah item A dan B terjadi bersamaan dalam transaksi yang sama dibagi dengan jumlah total dari transaksi tersebut. *Support* dapat dirumuskan sebagai berikut [4]:

$$Support = P\,(A \cap B)$$

$$= \frac{\text{Jumlah transaksi yang memuat A dan B}}{\text{Total jumlah transaksi}}$$

2. *Confidence* : *Confidence* dihasilkan dari seberapa kuat hubungan produk yang sudah dibeli.

Confidence dapat dirumuskan sebagai berikut [4] :

$$Confidence = P\,(B / A)$$

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

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 *interestingrule* atau *strongrule*. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap [19] :

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.

$$Support(A) = \frac{Jumlahtransaksi \text{ yang } mengandungA}{Jumlah\ totaltransaksi} \text{ X 100 \% ...Persamaan (2-1)}$$

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

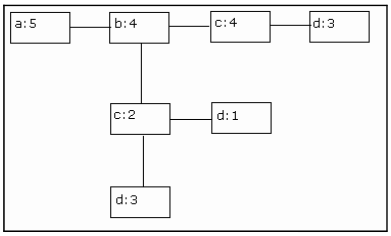
3. *Pattern Extraction*

   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 2.5). 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 horisontal 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*.

2.2.7 *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.