BAB 2
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Humblezing Company berdiri pada tahun 2010, Humblezing Company beralamat di Jl. Raya Cibereum No. 63 Cimahi Jawa Barat. Humblezing Company merupakan salah satu industri kreatif dibidang *fashion brand* atau biasa disebut dengan distro. Humblezing Company mendesain dan mengolah produknya sendiri. Mengawali usahanya dari hanya menerima pesanan sampai akhirnya karena ketekunan dan sifat kompetitif sang pemilik(*owner)* merekapun bisa berinovasi membuat produk sendiri tentu itu adalah sebuah pencapaian yang luar biasa bagi Humblezing.

Teamwork yang baik, kerja keras, terus berinovasi serta jujur lalu didasari filosofi dengan selalu memberikan kualitas dan pelayanan terbaik, humblezing company semakin mendapatkan kepercayaan dari konsumennya. Demi memberikan pelayanan terbaik humblezing company selalu aktif di media sosial, tanpa media sosial humblezing tidak akan mungkin bisa sampai memasarkan produk mereka ke setiap kota di Indonesia.

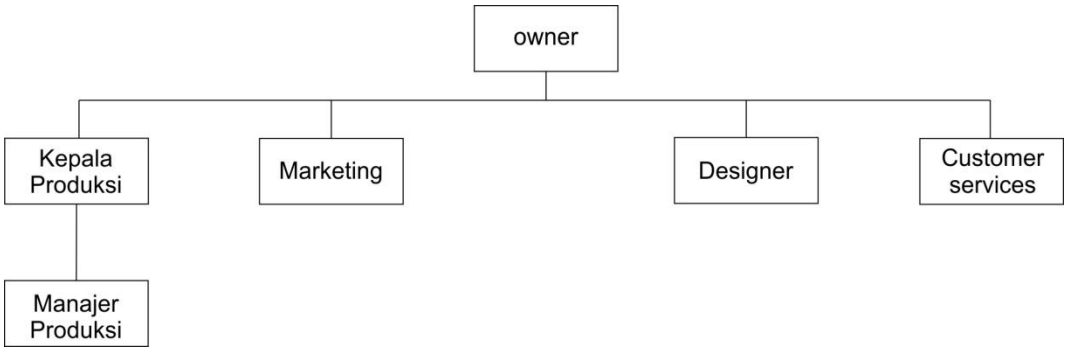
Karena kualitas produk yang baik dan bagus, Humblezing Company banyak menerima masukan dari konsumen untuk menambah jenis produk yang di produksi. seperti kemeja, jaket, dan baju dengan berbagai macam desain. Oleh karena itu Humblezing Company mampu bersaing dalam bisnis ini hingga sekarang.

2.1.1 Visi dan Misi Perusahaan

Humblezing Company miliki visi menjadi panutan industri *fashion brand* lain dalam bisnis *online*. Misi dari Humblezing Company sendiri adalah mengutamakan pelayanan demi kepuasan dan kenyamanan konsumen, serta selalu membuat produk kualitas terbaik.

2.1.2 Struktur Organisasi Perusahaan

Struktur organisasi perusahaan adalah gambar suatu bagan yang menerangkan posisi dan hierarki struktur kerja pegawai di dalam suatu perusahaan. Untuk struktur organisasi Humblezing Company dapat dilihat pada gambar dibawah ini:



Gambar 2. 1 Struktur Organisasi Humblezing Company

2.1.3 Deskripsi Kerja

a. *Owner*

   Owner bekerja sebagai orang yang bertanggung jawab atas semua tindakan yang diperlukan agar Humblezing Company berjalan lancar, menjalankan program perusahaan untuk mengejar target penjualan.

b. Kepala Produksi

   Kepala Produksi bertugas untuk mengontrol kinerja para pekerja, bertugas untuk melaporkan keadaan toko, mulai dari transaksi, keluhan konsumen serta masalah yang ada di toko. Selain itu kepala produksi juga bertugas menentukan bahan-bahan yang dipakai untuk proses produksi dari desain yang baru.

c. Manajer Produksi

   Manajer Produksi bekerja membantu kepala produksi dalam hal mengatur proses produksi, jenis produk mana yang harus lebih dulu diutamakan dalam proses pembuatannya.

d. Marketing

Marketing bekerja memasarkan produk secara *online* maupun *offline*, selain itu juga menganalisa dan mengembangkan strategi marketing, melakukan evaluasi untuk memastikan tercapainya target penjualan.

e. *Designer*

Designer yang membuat desain-desain dari setiap produk yang akan diproduksi. Biasanya designer sendiri di tuntut untuk membuat desain yang cukup banyak sehingga kemudian desain-desain tersebut dipilih oleh *owner* untuk di produksi

f. *Customer Services*

CS (*Customer Services*) bekerja melayani konsumen dari yang datang ke toko maupun secara online dan memberikan informasi terhadap produk yang konsumen minati.

2.2 Landasan Teori

Landasan teori membahas mengenai materi atau teori apa saja yang digunakan sebagai acuan dalam membuat tugas akhir ini. Landasan teori yang diuraikan merupakan hasil studi literatur, baik dari buku, maupun situs internet.

2.2.1 Data

Data adalah fakta atau apapun yang dapat digunakan sebagai input dan menghasilkan informasi[**6**]. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan kenyataan. Data merupakan suatu istilah yang berbentuk jamak dari kata "*datum*" yang berarti fakta atau bagian dari fakta yang mengandung arti yang menghubungkan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata, angka-angka, huruf-huruf yang menunjukan suatu ide, objek, kondisi dan situasi.

Menurut *the liang gie*: "Data atau bahan keterangan adalah hal atau peristiwa kenyataan lainnya apapun yang mendukung suatu pengetahuan untuk dijadikan dasar guna penyusunan keterangan pembuatan kesimpulan atau penetapan keputusan, atau

data ibarat bahan mentah yang melalui pengolahannya tertentu lalu menjadi keterangan (informasi)".

Kumpulan data yang saling berkaitan, berhubungan yang disimpan secara bersama-sama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan disebut basis data (*database*). Data-data ini harus
mengandung semua informasi untuk mendukung semua kebutuhan sistem. Proses dasar yang dimiliki oleh *database* ada empat, yaitu:

1. Pembuatan data-data baru (*create database*)
2. Penambahan data (*insert*)
3. Mengubah data (*update*)
4. Menghapus data (*delete*)

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna. *Database* menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan update menjadi rumit. Jadi tujuan dari pengaturan data dengan menggunakan *database* adalah :

a. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.
b. Cara pemasukan data sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
c. Pengendalian data untuk setiap siklus agar data selalu *up-to-date* dan dapat mencerminakan perubahan spesifik yang terjadi di setiap sistem.
d. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian dan gangguan-gangguan lain.

Dalam basis data sistem informasi digambarkan dalam model *entity relationship* (E-R). Bahasa yang digunakan dalam basis data (*database*) yaitu :

a. DDL (*Data Definition Language*)

   Merupakan bahasa definisi data yang digunakan untuk membuat dan mengelola objek *database* seperti *database*, tabel dan *view*.

b. DML (*Data Manipulation Language*)

   Merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data pada objek *database* seperti tabel.

c. DCL (*Data Control Language*)

   Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data.Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data, dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System* / DBMS).

2.2.2 Informasi

Informasi adalah hasil analisis dan sintesis terhadap data. Dengan kata lain, informasi dapat dikatakan sebagai data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang.

Menurut *Encyclopedia of Computer Science and Engineering*, banyak ilmuwan di bidang informasi menerima definisi standar bahwa informasi adalah data yang digunakan dalam pengambilan keputusan. Alasanya adalah bahwa informasi bersifat relatif, relatif terhadap situasi, relatif terhadap waktu saat keputusan diambil, juga relatif terhadap pembuat keputusan, dan bahkan juga relatif terhadap latar belakang pengambil keputusan.

Segala sesuatu yang dianggap penting pada suatu waktu bisa saja tidak berguna pada waktu yang lain. Ada kemungkinan pula bahwa sesuatu yang dianggap penting oleh seorang pengambil keputusan tidak dianggap penting bagi orang lain.

Dalam era reformasi, informasi menjadi sumber penting untuk melakukan pengambilan keputusan. Informasi dapat mengurangi ketidakpastian dan mempermudah pengambilan keputusan.

2.2.3 *Database Management System* (DBMS)

"*Managemen Sistem Basis Data (Database Management System / DBMS) adalah perangkat lunak yang di desain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar*".

Sistem Manajemen Basis data (*Database Management System*) merupakan sistem pengoperasian dan sejumlah data pada komputer. Dengan sistem ini dapat merubah data, memperbaiki data yang salah dan menghapus data yang tidak dapat dipakai. Sistem manajemen *database* merupakan suatu perluasan *software* sebelumnya mengenai *software* pada generasi komputer yang pertama. Dalam hal ini data dan informasi merupakan kesatuan yang saling berhubungan dan berkerja sama yang terdiri dari: peralatan, tenaga pelaksana dan prosedur data. Sehingga pengolahan data ini membentuk sistem pengolahan data. Peralatan dalam hal ini berupa perangakat keras (*hardware*) yang digunakan, dan prosedur data yaitu berupa perangakat lunak (*software*) yang digunakan dan dipakai untuk mengalokasikan dalam pembuatan sistem informasi pengolahan *database*.

Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data, serta pembuatan report dari data. Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi *user*. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisiensi yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas stuktur data. Sistem manajemen *database* atau *database management system* (DBMS) adalah merupakan suatu sistem *software* yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. *Database* sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki beberapa arti yang saling berpautan. DBMS yang utuh biasanya terdiri dari :

1. *Hardware*

   Hardware merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses *database*. Dalam sebuah organisasi berskala besar, *hardware* terdiri dari: jaringan dengan sebuah server pusat dan beberapa program *client* yang berjalan di komputer *desktop*.

2. *Software* beserta *utility Software* adalah DBMS yang aktual. DBMS memungkinkan para *user* untuk berkomunikasi dengan *database*. Dengan kata lain DBMS merupakan mediator antara *database* dengan *user*. Sebuah *database* harus memuat seluruh data yang diperlukan oleh sebuah organisasi.

3. Prosedur

   Bagian integral dari setiap sistem adalah sekumpulan prosedur yang mengontrol jalannya sistem, yaitu praktik-praktik nyata yang harus diikuti *user* untuk mendapatkan, memasukkan, menjaga, dan mengambil data.

4. Data

   Data adalah jantung dari DBMS. Ada dua jenis data. Pertama, adalah kumpulan informasi yang diperlukan oleh suatu organisasi. Jenis data kedua adalah metadata, yaitu informasi mengenai *database*.

5. Pengguna (*User*)

   Ada sejumlah user yang dapat mengakses atau mengambil data sesuai dengan kebutuhan penggunaan aplikasi-aplikasi dan *interface* yang disediakan oleh DBMS, antara lain adalah :

   a. *Database administrator* adalah orang atau *group* yang bertanggungjawab mengimplementasikan sistem *database* di dalam suatu organisasi.

   b. *Enduser* adalah orang yang berada di depan *workstation* dan berinteraksi secara langsung dengan sistem.

   c. *Programmer* aplikasi, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

2.2.3.1 Data Pada database dan Hubungannya

Ada 3 jenis data pada sistem *database*, yaitu[**7**]:

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.

2.2.3.2 Keuntungan dan Kerugian Pemakaian Sistem Database

Keuntungan[**7**]:

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/kompleks.
3. Penganan proses *recovery backup* sulit.
4. Kerusakan pada sistem basis data dapat mempengaruhi.

2.2.4 *Data Mining*

Data Mining adalah proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar. *Data Mining* juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan[**6**].

Data Mining adalah salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari *database* skala besar yang makin banyak terakumulasi sejalan dengan pertumbuhan teknologi informasi. Defisini umum dari *data mining* itu sendiri adalah serangkaian proses untuk menggali nilai tambah berupa pengatahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data[**8**].

Data mining terutama digunakan untuk mencari pengetahuan yang terdapat dalam database yang besar sehingga sering disebut *Knowledge Discovery in Databases* (KDD). Proses pencarian pengetahuan ini menggunakan berbagai teknik-teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksikannya. Proses pencarian bersifat iteratif dan interaktif untuk menemukan pola atau model yang sahih, baru, bermanfaat, dan dimengerti.

2.2.4.1 Metode-Metode Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu[**11**]:

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 :

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

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 :

   a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
   b. 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 :

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk sebuah perusahaan yang tidak memiliki dana pemasaran yang besar.
b. 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 :

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

2.2.4.2 *Cross-Industry Standard Process For Data Mining* (CRISP-DM)

Metode penelitian *data mining* yang dipakai adalah CRISP-DM (*CRoss-Industry Standard Process for Data Mining*)[**1**]. CRISP-DM ini merupakan suatu konsorsium perusahaan yang didirikan oleh Komisi Eropa pada 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*. Gambar berikut menjelaskan tentang siklus hidup pengembangan *data mining* yang telah ditetapkan dalam CRISP-DM[**4**].



Gambar 2. 2 Metode *CRISP-DM*

Berikut ini adalah penjelasan mengenai enam tahap siklus hidup pengembangan *data mining* berdasarkan gambar di atas :

1. *Business Understanding*

   Tahap pertama adalah memahami tujuan dan kebutuhan dari sudut pandang bisnis, kemudian menterjemakan pengetahuan ini ke dalam pendefinisian masalah dalam *data mining*. Selanjutnya akan ditentukan rencana dan strategi untuk mencapai tujuan tersebut.

2. *Data Understanding*

Tahap ini dimulai dengan pengumpulan data yang kemudian akan dilanjutkan dengan proses untuk mendapatkan pemahaman yang mendalam tentang data, mengidentifikasi masalah kualitas data, atau untuk mendeteksi adanya bagian yang menarik dari data yang dapat digunakan untuk hipotesa untuk informasi yang tersembunyi.

3. *Data Preparation*

Tahap ini meliputi semua kegiatan untuk membangun dataset akhir (data yang akan diproses pada tahap pemodelan/*modeling*) dari data mentah. Tahap ini dapat diulang beberapa kali. Pada tahap ini juga mencakup pemilihan tabel, *record*, dan atribut-atribut data, termasuk proses pembersihan dan transformasi data untuk kemudian dijadikan masukan dalam tahap pemodelan (*modeling*).

4. *Modeling*

Dalam tahap ini akan dilakukan pemilihan dan penerapan berbagai teknik pemodelan dan beberapa parameternya akan disesuaikan untuk mendapatkan nilai yang optimal. Secara khusus, ada beberapa teknik berbeda yang dapat diterapkan untuk masalah *data mining* yang sama. Di pihak lain ada teknik pemodelan yang membutuhan format data khusus. Sehingga pada tahap ini masih memungkinan kembali ke tahap sebelumnya.

5. *Evaluation*

Pada tahap ini, model sudah terbentuk dan diharapkan memiliki kualitas baik jika dilihat dari sudut pandang analisa data. Pada tahap ini akan dilakukan evaluasi terhadap keefektifan dan kualitas model sebelum digunakan dan menentukan apakah model dapat mencapat tujuan yang ditetapkan pada fase awal (*Business Understanding*). Kunci dari tahap ini adalah menentukan apakah ada masalah bisnis yang belum dipertimbangkan. Di akhir dari tahap ini harus ditentukan penggunaan hasil proses *data mining*.

6. *Deployment*

Pada tahap ini, pengetahuan atau informasi yang telah diperoleh akan diatur dan dipresentasikan dalam bentuk khusus sehingga dapat digunakan oleh pengguna. Tahap deployment dapat berupa pembuatan laporan sederhana atau mengimplementasikan proses *data mining* yang berulang dalam perusahaan. Dalam banyak kasus, tahap *deployment* melibatkan konsumen, di samping analis data, karena sangat penting bagi konsumen untuk memahami tindakan apa yang harus dilakukan untuk menggunakan model yang telah dibuat.

2.2.5 *Association Rule*

Assocation rule adalah salah satu teknik utama atau prosedur dalam *Market Basket Analysis* untuk mencari hubungan antar-item dalam suatu data-set dan menampilkan bentuk *association rule*[**2**]. *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 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 assosiatif atau aturan keterkaitan yang memenuhi syarat yang telah ditentukan[**9**].

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 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 ditujukkan sebagai berikut :

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

Association rule diperlukan 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% menunjukkan bila konsumen membeli kopi dan pasti membeli susu sebesar 60%. *Association rule* memiliki dua tahap pengerjaan, yaitu:

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset.*
2. Mendefinisikan *condition* dan *result* (untuk *conditional association 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. 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[**2**] :

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.

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

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

$$Support(A, B) = P(A \cap B) \quad \text{Persamaan (2-2)}$$

$$Support(A, B) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ total\ transaksi} \quad \text{Persamaan (2-3)}$$

2. Pembentukan aturan asosiasi

   Setelah semua pola frekuensi tinggi ditemukan, kemudian mencari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A -> B dari *support* pola frekuensi tinggi A dan B, menggunakan rumus :

$$Confidence = P(B \mid A) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ transaksi\ yang\ mengandung\ A} \quad \text{Persamaan (2-4)}$$

2.2.6 Algoritma *FP-Growth*

Algoritma *FP-Growth* merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algortima Apriori diperbaiki oleh algoritma *FP-Growth*.

Frequent Pattern Growth (*FP-Growth*) adalah salah satu alternatif algoritma yang digunakan untuk menentukan himpunan data yang paling sering muncul dalam sebuah kumpulan data.

Pada algortima Apriori diperlukan *generate candidate* untuk mendapatkan *frequent itemset*. Akan tetapi, pada algortima *FP-Growth generate candidate* tidak

dilakukan karena *FP-Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset*. Hal ini yang menyebabkan algortima *FP-Growth* lebih cepat dari algortima Apriori.

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algortima *FP-Growth* dapat langsung mengekstrak *frequent itemset* dari *FP-Tree*.

Penggalian *itemset* yang *frequent* dengan menggunakan algortima *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *FP-Tree*. Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu[**12**]:

a. Tahap pembangkitan *Conditional pattern base*.
b. Tahap pembangkitan *conditional FP-Tree*
c. Tahap pencarian *frequent itemset*

Algoritma *FP-Growth* menggunakan struktur data *FP-Tree*. Informasi yang disimpan sebuah *node FP-Tree*: *Item, Index parent, Support,* dan *Next* (*Pointer*). Ketika selesai membuat *FP-Tree*, kita tidak begitu saja bisa mendapatkan *frequent itemset* yang terdapat dalam *dataset*. Suatu kombinasi *itemset* bisa saja berada di beberapa *path* yang berbeda. Untuk mendapatkan suatu *pattern* dalam *FP-Growth* langkah yang lebih mudah adalah mencari arah dari ujung suatu *path*, kemudian kita mencari mulai dari *header* untuk *item* di ujung tersebut, barulah kemudian dibuat berdasarkan tiap *node* berisi *item* tersebut dicari arah *path node* ke atas. Hal ini tentu lebih cepat dari pada *up-down* karena *pointer* langsung yang dimiliki tiap *node* adalah *pointer* ke *parent*. *Path* yang dieksplorasi hanyalah *path-path* yang memiliki *node* yang sedang dicari. Jadi dalam struktur *FP-Tree* ada *link* dari suatu *item* ke *path-path* yang memiliki item tersebut, sehingga ketika dibutuhkan pencarian *pattern-pattern* untuk suatu item tertentu, hanya mencari dari *path-path* tersebut saja. Berikut adalah tahapan-tahapan dari algoritma *FP-Growth*[**13**]:

1. Membuat *Harder 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.

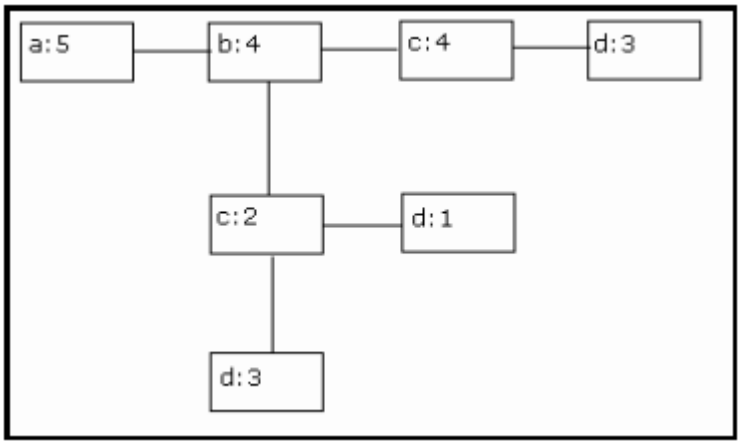
3. *Pattern Extraction*

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 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

PatternTree. Setiap *node* di *PatternTree* merepresentasikan dan menyimpan frekuensi suatu *pattern*. *PatternTree* terdiri atas *PatternTreeNode* yang menyimpan nilai *item*, nilai *support* dan dilengkapi dengan dua *pointer* yaitu untuk horisontal dan vertikal.



Gambar 2. 3 *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:

a. a:5 menggambarkan bahwa ada *Pattern* a sebanyak 5
b. b:4 menggambarkan bahwa ada *Pattern* a-b sebanyak 4
c. c:4 menggambarkan bahwa ada *Pattern* a-b-c sebanyak 4
d. d:3 menggambarkan bahwa ada *Pattern* a-b-c-d sebanyak 3
e. c:2 menggambarkan bahwa ada *Pattern* a-c sebanyak 2
f. d:1 menggambarkan bahwa ada *Pattern* a-c-d sebanyak 1
g. 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*.

Struktur data-struktur data yang digunakan dalam algoritma *FP-Growth* adalah *FP-Tree* yang tersusun dari *FPTNode, PatternTree* yang terdiri dari *PatternTreeNode,*dan juga *FItemset*. *FP-Tree* adalah struktur *tree* yang menyimpan item-item yang telah ditemukan pada saat membaca data. Kemudian *FP-Tree* ini terdiri atas *FPTNode* yang membentuk suatu *path* untuk dicari terdapat *pattern* apa saja pada *path* tersebut. *FPTNode* adalah *node* pada *FP-Tree* yang menyimpan informasi *item* pada *node,* nilai *support*, *pointer* ke *parent*, dan *pointer* ke *node* berikutnya yang mempunyai item yang sama (*next*). *PatternTree* adalah struktur untuk menyimpan semua *pattern* yang ditemukan pada *FP-Tree*. *FItemset* menyimpan *frequent itemset* yang telah ditemukan. Format ini digunakan terutama bila ingin menghasilkan *rule*, untuk membandingkan suatu *frequent itemset* dengan subset-nya, akan lebih mudah dengan struktur seperti ini daripada mengolah *frequent itemset* masih dalam bentuk *tree*.

2.2.7 DFD (*Data Flow Diagram*)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

a. Kebebasan dari menjalankan implementasi dalam teknis sebuah sistem.
b. Pemahaman lebih jauh mengenai keterkaitan satu dama lain dalam sistem dan subsistem.

c. Mengkomunikasikan pengetahuan sistem yang ada dengan *user* melalui diagram aliran data.
d. Menganalisa sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Selain kelebihan utama DFD sendiri memiliki kelebihan tambahan lainnya, yaitu sebagai berikut:

1. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan *user*.
2. Memungkinkan pengalisis menggambarkan setiap komponen yang digunakan dalam diagram.
3. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
4. Dapat digunakan sebagai latihan yang bermanfaat bagi pengalisis, sehingga memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.

DFD terdiri dari *context* diagram dan diagram rinci (DFD *Levelled*). *Context* diagram berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. DFD *levelled* menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Dalam DFD *levelled* akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam DFD *levelled* bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut. Proses yang tidak dapat

diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

2.2.8 Kamus Data (*Data Dictionary*)

Merupakan tempat penyimpanan dari elemen-elemen yang berada dalam satu sistem. Kamus data mempunyai fungsi yang sama dalam pemodelan sistem dan juga berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail, dan me-reorganisasi semua elemen data yang digunakan dalam sistem sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

2.2.9 MySQL

SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server database*. Semenjak tahun 70-an bahasa ini telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya dBase ataupun Clipper yang masih menggunakan perintah-perintah pemrograman murni.

MySQL adalah sebuah server *database* SQL *multiuser* dan *multi-threaded*. SQL sendiri adalah salah satu bahasa database yang paling populer di dunia. Implementasi program server database ini adalah program daemon 'mysqld' dan beberapa program lain serta beberapa pustaka. Sebagaimana *database* sistem yang lain, dalam SQL juga dikenal hierarki server dengan database-database. Tiap-tiap *database* memiliki tabel-tabel. Tiap-tiap tabel memiliki *field-field*. Umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur 2 dimensi terdiri atas baris dan kolom. *Field-field* tersebut dapat berupa data seperti int, *realm char*, *date*, *time* dan lainnya. SQL tidak memiliki fasilitas pemrograman yang lengkap, tidak ada *looping* ataupun percabangan. Sehingga untuk menutupi

kelemahan ini perlu digabung dengan bahasa pemrograman semisal Pascal. MySQL sering digunakan sebagai SQL server karena berbagai kelebihannya, antara lain :

1. Sintaksnya lebih mudah dipahami dan tidak terlalu rumit.
2. Source MySQL dapat diperoleh dengan mudah dan gratis.
3. Pengaksesan *database* dapat dilakukan dengan mudah.

2.2.10 Microsoft Visual Studio .Net dan C# (*Sharp*)

Visual studio .Net adalah sekumpulan alat pengembangan *software* yang diperuntukkan bagi .Net *platform*. Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft *Intermediate Language* di atas .NET *Framework*).

Bahasa C# adalah sebuah bahasa pemrograman yang bersifat *general-purpose*, berorientasi objek, yang dapat digunakan untuk membuat program di atas arsitektur Microsoft .NET *Framework*. Bahasa C# ini memiliki kemiripan dengan bahasa Java, C dan C++ (selengkapnya dapat dilihat pada Sejarah Bahasa C#).

Bahasa pemrograman ini dikembangkan oleh sebuah tim pengembang di Microsoft yang dipimpin oleh Anders Hejlsberg, seorang yang telah lama malang melintang di dunia pengembangan bahasa pemrograman karena memang ialah yang membuat Borland Turbo Pascal, Borland Delphi, dan juga Microsoft J++.

Kini, C# telah distandarisasi oleh *European Computer Manufacturer Association* (ECMA) dan juga *International Organization for Standardization* (ISO) dan telah menginjak versi 3.0 yang mendukung beberapa fitur baru semacam *Language Integrated Query* (LINQ) dan lain-lainnya