BAB 2
TINJAUAN PUSTAKA

2.1 Profil Institusi

2.1.1 Sejarah Rumah Makan Roso Echo

Rumah Makan Roso Echo merupakan rumah makan yang pertama dan satu-satunya rumah makan di Cirebon yang menyediakan aneka masakan jawa dengan kualitas rasa khas masakan jawa. Rumah Makan Roso Echo ini pertama kali didirikan di Jalan pangeran cakrabuana BTN Cempaka Arum A-3, Sumber Cirebon, pada tanggal 1 September 2006 oleh Ir. Suwarno. Selain menyediakan masakan khas jawa, Rumah Makan Roso Echo ini menerapkan konsep prasmanan dengan nuansa khas jawa dalam pelayanannya, sehingga memberikan kemudahan bagi para konsumen untuk melihat dan memilih secara langsung menu makanan yang sesuai dengan keinginan dan kebutuhannya tanpa harus menunggu menu yang disajikan. Selain itu, konsumen juga dapat menikmati nuansa jawa yang belum pernah konsumen dapatkan di rumah makan lain dan rumah makan ini menampilkan nuansa persawahan dengan pemandangan Gunung Ciremai yang indah. Hal inilah yang menjadikan rumah makan ini sangat dinikmati oleh para konsumen dari ebrbagai daerah yang ada di wilayah Cirebon.

Seiring dengan minat konsumen terhadap Rumah Makan Roso Echo yang semakin meningkat setiap tahunnya, sehingga pada tahun 2010 tepatnya tanggal 1 Juli 2010, Rumah Makan Roso Echo ini resmi membuka cabang yang berlokasi tak jauh dari pusatnya, yaitu di Jalan Sultan Ageng Tirtayasa No. 4, Sumber, Cirebon. Kini cabang Rumah Makan Roso Echo yang didirikan pada tahun 2010 telah mengalami perkembangan yang semakin pesat. Dahulu rumah makan ini hanya memiliki satu ruangan bagi konsumen untuk menikmati makanannya, dan kini rumah makan ini telah memiliki beberapa ruangan yang dapat konsumen gunakan.

2.1.2 Logo

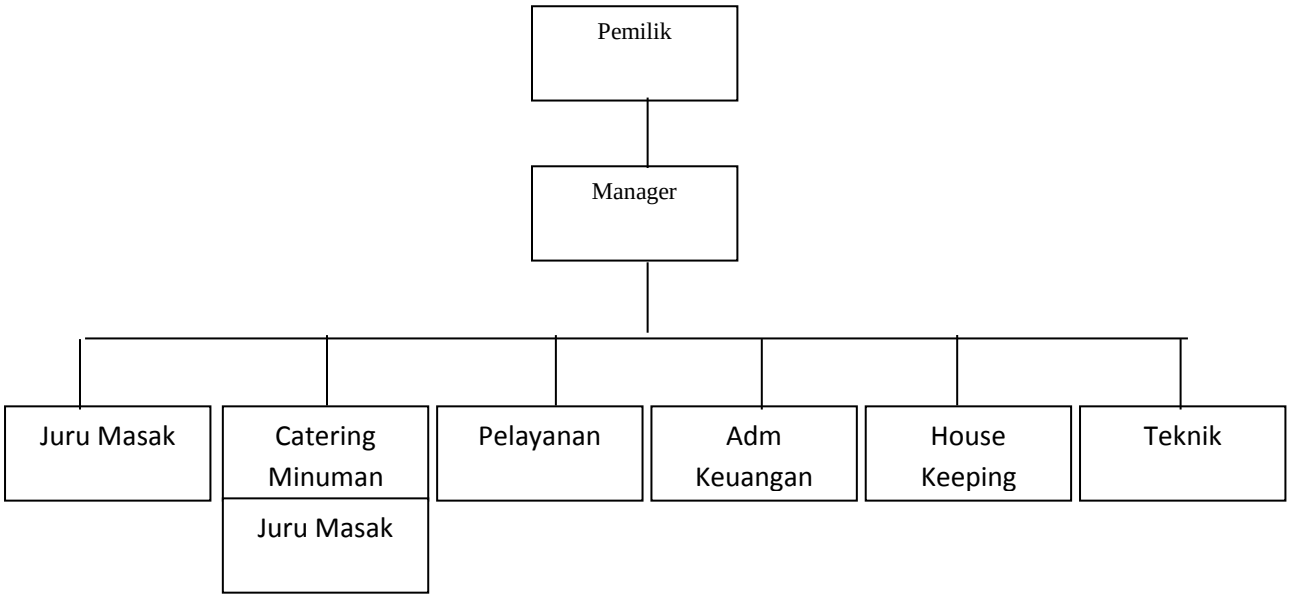
Berikut ini adalah logo dari rumah makan Roso Echo yang dapat dilihat pada Gambar 2.1 :



Gambar 2. 1 Logo Rumah Makan Roso Echo

2.1.3 Struktur Organisasi

Berikut ini adalah struktur organisasi rumah makan Roso Echo yang dapat diliihat pada Gambar 2.2 :



Gambar 2. 2 Struktur organisasi rumah makan Roso Echo

Job description :

1. Pemilik bertanggung jawab penuh terhadap rumah makan, tugas dan tanggung jawabnya sebagai berikut :
   a. Membuat perencanaan, strategi dan kebijakan yang menyangkut operasi rumah makan.
   b. Menyusun anggaran kebutuhan persediaan bahan baku.
   c. Melakukan kontrol secara keseluruhan atau operasi rumah makan.
   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 rumah makan.
   b. Mengatur setiap bagian yang ada di rumah makan agar menjalankan tugasnya dengan baik.
3. Bagian juru masak memiliki tugas dan tanggung jawab sebagai berikut:
   a. Memasak semua menu yang harus disediakan di meja makan prasmanan.
   b. Bertanggung jawab atas pesanan dari pelanggan.
   c. Bertanggung jawab atas hasil masakan.
   d. Mengontrol persediaan bahan baku dari menu makanan.
4. Bagian catering dan minuman memiliki tugas dan tanggung jawab sebagai berikut :
   a. Menyiapkan catering pesanan dari pelanggan.
   b. Mengantarkan pesanaan catering kepada pelanggan.
   c. Menyiapkan minuman yang dipesan oleh tiap pelanggan.
   d. Mengontrol persediaan bahan baku minnuman.
5. Bagian pelayanan berfungsi untuk melayani tiap pelanggan, tugas dan tanggung jawab sebagai berikut :
   a. Membantu dalam melayani pesanan-pesanan pelanggan.
   b. Bertanggung jawab atas kebutuhan piring, sendok, gelas, menu makanan, dan lain-lain.
   c. Melayani semua pesaan pelanggan.

6. 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.
7. 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 rumah makan.
   b. Membereskan dan membersihkan peralatan makan yang telah pelanggan gunakan.
8. Bagian teknik bertugas dan bertanggung jawab terhadap semua keperluan teknisi yang ada di rumah makan.

2.1.4 Visi dan Misi

2.1.4.1 Visi

Visi dari rumah makan Roso Echo adalah sebagai berikut :

1. Mendirikan rumah makan dengan suasana konsep alam.
2. Mewujudkan makanan berkualitas, dengan cita rasa makanan tradisional.

2.1.4.2 Misi

Misi dari rumah makan Roso Echo adalah sebagai berikut :

1. Mengutamakan layanan serta kualitas makanan yang akan disajikan.
2. Berinovasi secara berkala, baik dalam produk maupun sistem penjulan yang diterapkan.
3. Mampu bersaing dengan rumah makan lain yang ada di dalam ataupun di luar wilayah Cirebon.
4. Mengembangkan usaha rumah makan di luar wilayah Cirebon.

2.2 Landasan Teori

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

2.2.1 Data

Data adalah segala fakta, angka atau teks yang dapat diproses oleh komputer. Data dapat digunakan sebagai *input* dan menghasilkan sebuah informasi. Data adalah sesuatu yang belum memiliki arti dan masih membutuhkan suatu pengolahan. Dalam data terdapat himpunan data yang merupakan kumpulan dari objek dan atributnya. Atribut merupakan sifat atau karakteristik dari suatu objek yang biasanya dikenal sebagai *variabel, field,* karakteristik atau fitur.Salah satu himpunan data adalah *record* data, yaitu data yang terdiri dari sekumpulan *record*, yang masing-masing terdiri dari satu set atribut yang tetap. Salah satu yang termasuk dalam tipe data *record* yaitu data transaksi. Data transaksi merupakan sebuah tipe khusus dari *record* data, dimana tiap *record* (transaksi) meliputi satu *set item* [2].

2.2.2 *Database*

Database adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memEnuhi informasi yang optimal yang dibutuhkan oleh para pengguna [3].

2.2.2.1 Data pada Database dan Hubungannya

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

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 [3] :

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

Kerugian :

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

2.2.3 Data Mining

Data mining adalah proses untuk mendapatkan informasi yang berguna dari gedung basis data yang besar. *Data mining* juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah *data mining* kadang disebut juga *knowledge discovery*. Salah satu teknik yang dibuat dalam *data mining* adalah bagaimana menelusuri data yang ada untuk membangun sebuah model, kemudian menggunakan model tersebut agar dapat mengenali pola data yang lain yang tidak berada dalam basis data yang tersimpan. Dalam *data mining*, pengelompokan data juga bisa dilakukan. Tujuannya adalah agar kita dapat megetahui pola universal data-data yang ada [4].

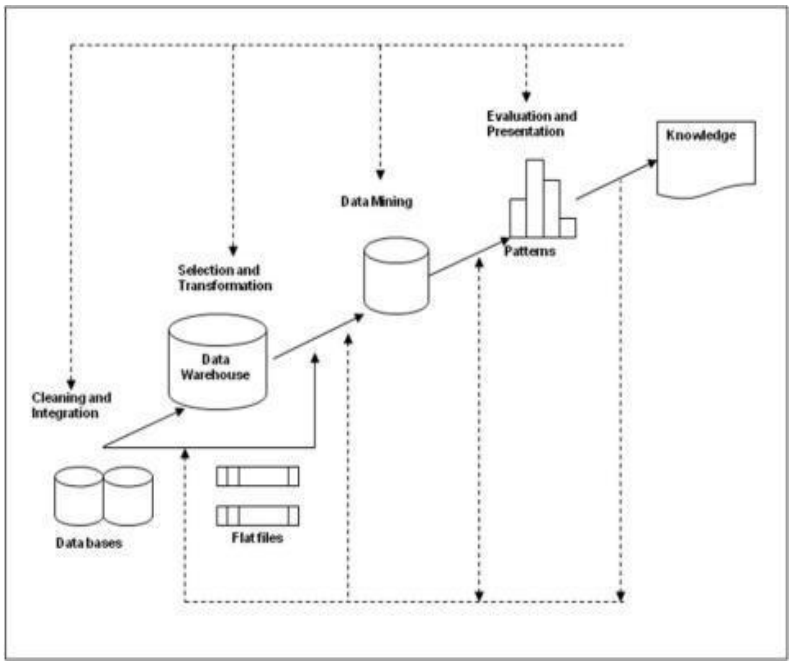
Istilah *data mining* sering disebut dengan nama *knowledge discovery atau* penemuan pengetahuan karena tujuan utama dari *data mining* untuk mendapatkan pengetahuan yang masih tersembunyi di dalam bongkahan data. Secara umum, definisi *data mining* dapat diartikan sebagai berikut[5]:

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.

2.2.3.1 Konsep *Data Mining*

Data mining sangat diperlukan terutama dalam mengelola data yang sangat besar untuk meudahkan aktifitas recording suatu transaksi dan untuk proses *data warehousing* agar dapat memberikan informasi yang akurat bagi pengguna *data mining*. Alasan utama *data mining* sangat dibutuhkan dalam industri informasi karena tersedianya data dalam jumlah yang besar dan semakin besarnya kebutuhan untuk mengubah data tersebut menjadi informasi dan pengetahuan yang berguna karena sesuai fokus bidang ilmu ini yaitu melakukan kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran atau berjumlah besar. Informasi iniliah yang nantinya sangat berguna untuk pengembangan. Berikut adalah langkah-langkah dalam *data mining* [5] :



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*.

2.2.3.2 Tahapan *Data Mining*

Tahapan dalam melakukan *data mining* salah satunya adalah *preprocessing* data. Tahapan ini biasanya diperlukan karena data yang akan digunakan belum baik, yang disebabkan oleh beberapa faktor berikut ini [6]:

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

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

a. Akurat
b. Lengkap
c. Konsisten
d. Relevan
e. Bisa dipercaya

f. Mempunyai nilai tambah

g. Kemudahan untuk dimengerti

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

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

2.2.3.3 Metode –Metode *Data Mining*

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu [7] :

1. Deskripsi

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

2. Estimasi

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

3. Prediksi

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

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

4. Klasifikasi

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

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

5. Pengklusteran

Pengklusteran merupakan pengelompokan record, pengamatan, atau memperhatikan dan membentuk 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

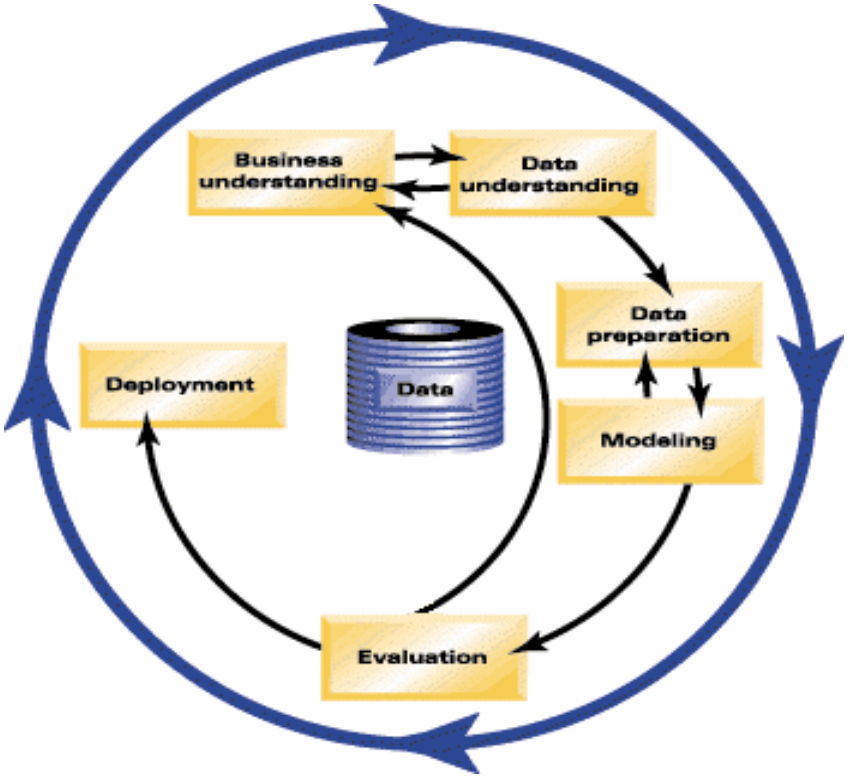
Tugas asosiasi dalam data mining adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang pasar. Contoh asosiasi dalam bisnis dan penelitian adalah :

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.

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

Cross-Indutry Standard Prosess for Data Mining (CRISP-DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industri seperti Daimler Chrysler, SPSS dan NCR. CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP-DM, sebuah proyek *data mining* memiliki siklus hidup yang

terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan antarfase digambarkan dengan panah. Sebagai contoh, jika proses berada pada fase *modelling*. Berdasar pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuk perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation*. Enam fase yang ada dalam metode CRISP-DM tersebut dapat dilihat pada Gambar 2.4 :



Gambar 2. 4 CRISP-DM

Fase-fase dari CRISP-DM [8] :

1. *Business understanding*
   a. Penentuan tujuan objek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
   b. Menerjemahkan tujuan dan batasan menjadi formula dari permasalahan *data mining*.

2. *Data understanding*
   a. Mengumpulkan data.
   b. Menggunakan analisis penyelidikan data untuk mengenali lebih lanjut data dan pencarian pengetahuan awal.
   c. Mengevaluasi kualitas data.
   d. Jika diinginkan, pilih sebagian kecil grup data yang mungkin mengandung pola dari permasalahan.
3. *Data preparation*
   a. Siapkan dari data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini merupakan pekerjaan berat yang perlu dilaksanakan secara intensif.
   b. Pilih kasus dan variabel yang ingin dianalisis dan yang sesuai dengan analisis yang akan dilakukan.
   c. Lakukan perubahan pada beberapa variabel jika dibutuhkan.
   d. Sipakan data awal sehingga siap untuk perangkat pemodelan.
4. *Modelling*
   a. Pilihan dan aplikasikan teknik pemodelan yang sesuai.
   b. Kalibrasi aturan model untuk mengoptimalkan hasil.
   c. Perlu diperhatikan bahwa beberapa teknik mungkin untuk digunakan pada permasalahan *data mining* yang sama.
   d. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu.
5. *Evaluation*
   a. Mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan.
   b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.
   c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.

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

6. *Deployment*

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

b. Contoh sederhana penyebaran : pembuataan laporan.

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

2.2.4 *Association Rule*

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

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

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

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

Association rule dperlukan suatu variabel ukuran yang ditentukan sendiri oleh *user* untuk menentukan batasan sejauh mana atau sebanyak apa *output* yang diinginkan *user*. *Support* dan *confidence* adalah sebuah ukuran kepercayaan dan kegunaan suatu pola yang telah ditemukan, Nilai *support* 2% menunjukkan bahwa

keseluruhan dari total transaksi konsumen membeli kopi dan susu secara bersamaan yaitu sebanyak 2%. Sedangkan *confidence* 60%, yaitu menunjukkan bila konsumen membeli kopi dan pasti membeli susu sebesar 60%.

Penting tidaknya suatu aturan asosiasif dapat diektahui dengan dua parameter yaitu *support* dan iconfidence. *Support* (nilai pnunjang) adalah persentase kombinasi *item* tersebut dalam *database*, sedangkan *confidencei* (nilai kepastian adalah kuatnya hubungan antar –*item* dalam aturan asosiasi.

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

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

Kedua ukuran ini nantinya berguna dalam menentukan kekuatan suatu pola dengan membandingkan pola tersebut dengan nilai minimum kedua parameter tersebut yang ditentukan oleh pengguna. Bila suatu pola memenuhi kedua nilai minimum parameter yang sudah ditentukan sebelumnya, maka pola tersebut dapat disebut sebagai *interesting rule* atau *strong rule*. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap [12] :

1. Analisis pola frekuensi tinggi

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

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

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

$Support(A, B) = P(A \cap B)$…Persamaan (2-2)

$Support(A, B) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ total\ transaksi}$ x100%.Persamaan(2-2)

2. Pembentukan aturan asosiasi

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

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

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

Jika A, maka B (A→B)

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

Jika A, maka B

Tidak berarti

Jika B, maka A

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

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

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

Contoh kasus penerapan Association Rule *Support* dan *Confidence*

- Misalkan didapatkan sebuah item dan frekuensi yang muncul daam transaksi penjualan sepeda motor .Sudah ditentuin sebelumnya bahwa item 1 sampai 5 adalah Beat, Mio, Yupiter, Supra X, Kawasaki Ninja. Jumlah tran saksinya adalah 5 data transaksi.

- Support count item 1 yang keluar sebelumnya yaitu 3, Support count item 2 yang keluar sebelumnya yaitu 4, Support count item 3 yang keluar sebelumnya yaitu 2, Support count item 4 yang keluar sebelumnya yaitu 5, Support count item 5 yang keluar sebelumnya yaitu 2

- Selanjutnya *Item* yang frequent yang keluar

Tabel 2. 1 Frequent yang Muncul

Item	***Frequent***
1	10
2	6
3	4
4	15
5	4

- Setelah itu frequent diurutkan sesuai dengan frequent terbesar sampai terkecil.

Tabel 2. 2 Setelah diurutkan

Item	***Frequent***
4	15
1	10
2	6
3	4
5	4

- Setelah mendapatkan hasil *frequent itemset* , proses selanjutnya adalah menentukan *frequent itemset* setiap suffix

Tabel 2. 3 Hasil *Frequent Itemset*

Suffix	***Frequent Itemset***
4	4, 4 –1
1	1, 1 - 3
2	2, 2 - 5
3	3, 3 - 2
5	5, 5 – 4

- Tentukan nilai *Minimum support* sebesar 15%, dan *confidence* 50%.

Tabel 2. 4 Hasil Perhitungan *Support* dan *Confidence*

Rule	***Support***	***Confidence***
4→ 1	5/5*100=100	5/15*100=33,33%
1→4	5/5*100=100	5/10*100=50%
1→ 3	3/5*100=60	3/10*100=30%
3→1	3/5*100=60	3/4*100=75%
2 →5	4/5*100=80	4/6*100=66,66%
5→2	4/5*100=80	4/4*100=100%
3 →2	2/5*100=40	2/4*100=50%
2→3	2/5*100=40	2/6*100=33,33%
5 →4	2/5*100=40	2/4*100=50%
4→5	2/5*10=40	2/15*100=13,33%

- Setelah melakukan perhitungan *minimum support* dan *confidence*, jika terdapat yang tidak memenuhi *support* dan *confidence* yang ditentukan sebelumnya makan akan dihapus.

Tabel 2. 5 Hasil *Generate Rule*

Item	***Support***	***Confidence***
1→4	5/5*100=100	5/10*100=50%
3→1	3/5*100=60	3/4*100=75%
2 →5	4/5*100=80	4/6*100=66,66%
5→2	4/5*100=80	4/4*100=100%
3 →2	2/5*100=40	2/4*100=50%
5 →4	2/5*100=40	2/4*100=50%

2.2.5 Algoritma FP-Growth

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

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

1. Tahap Pembangkitan *conditional pattern base.*

   Conditional Pattern Base merupakan sub *database* yang berisi *prefix path* dan *suffix pattern*. Pembangkitan *conditinal pattern base* didapatkan melalui FP-Tree yang telah dibangun sebelumnya.

2. Tahap Pembangkitan *Conditional* FP-Tree

   Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap item yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support count* ξ akan dibangkitkan dengan *conditional* FP-Tree.

3. Tahap Pencarian *frequent itemset*

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

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

```
Input       : FP-Tree Tree
Output      : Rt Sekumpulan lengkap pola frequent
Method      : FP-Ggrowth ( Tree, null )
Procedure   : FP-Growth ( Tree,  _ )
{
01 : if Tree mengandung single path P;
02 : then untuk tiap kombinasi ( dinotasikan _ ) dari node node dalam path do
03 : bangkitkan pola _ _  dengan support daro node-node dalam _;
04 : else untuk tiap a1 dalam header dari Tree
do
{
05 : bangkitkan pola
06 : bangun _ = a1 _ dengan support = a1.
Support
07 : if Tree _ = _
08 : then panggil FP-Growth  (Tree, _ )
}
```

Contoh kasus penerapan Algoritma *FP-Growth*

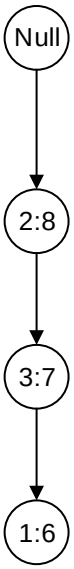
- Misalkan didapat hasil dari tabel pembangkitan *conditional pattern base* sebagai berikut, diketahui sebelumnya nilai minimum support 5%.
  a. Tahap Pembangkitan Conditional Pattern Base

Tabel 2. 6 *Conditional Pattern base*

Item	***Conditional Pattern base***
1	• 2,3, 1:6

  b. Tahap Pembangkitan *Conditional FP-Tree*
     Pada tahap ini, *support count* dari setiap item dijumlahkan, lalu setiap dan dibangkitkan dengan *conditional fp-tree*

- *Conditional FP-Tree* 1 : { 2, 4, 3}



Gambar 2. 5 *Conditional FP-Tree* 1

Karena *support count* 1 adalah 6, maka memenuhi *minimum support* yaitu 5. Berikut adalah tahap-tahap pembangkitan *conditional fp-tree* 1.

- Pada lintasan yang berakhiran {1} dengan *support count* 1 atau lebih yang dimilikinya akan dihilangkan dan *support count* yang melewati

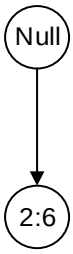
lintasan pertama atau lintasan lainnya yang berakhiran {1}, akn disamain sesuai dengan support count yang dimilikinya.

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

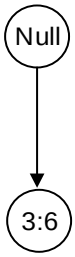


Gambar 2. 6 *Conditional FP-Tree* 1 Setelah Dibangkitkan

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



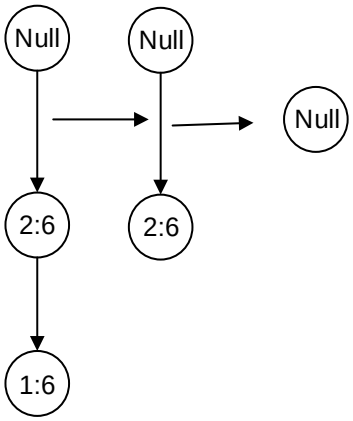
Gambar 2. 7 *Conditional FP-Tree* 1,2:6 Hasil *Frequent Itemset*



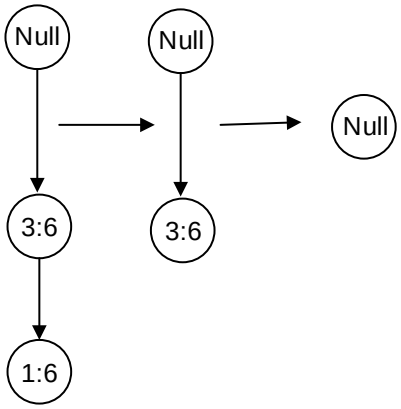
Gambar 2. 8 *Conditional FP-Tree* 1,3:6 Hasil *Frequent Itemset*

c. Tahap Pencarian *frequent itemset*

- Akumulasiakn count setiap item
- Buat *conditional fp-tree*



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



Gambar 2. 10 *Conditional FP-Tree* Untuk *Item Frequenset* 1, 3:6

2.2.6 FP-*Tree*

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

1. *Fp-tree* dibentuk oleh sebuah akar yang diberi *label null*, sekumpulan pohon yang beranggotakan item-item tertentu, dan sebuah tabel *frequent header*.

2. Setiap simpul dalam *fp-tree* mengandung tiga informasi penting, yaitu *label item*, menginformasikan jenis item yang dipresentasikan simpul tersebut, *support count* merepresentasikan jumlah lintasan transaksi yang melalui simpul tersebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label item sama antar lintasan, ditandai dengan garis putus-putus.

2.2.7 Alat-Alat Pemodelan Sistem

Alat-alat pemodelan sistem adalah alat dan metode yang akan digunakan untuk memodelkan perancangan perangkat lunak dalam skripsi ini.

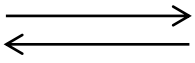
2.2.7.1 Diagram Konteks

Diagram konteks adalah diagram aliran data pada tingkat paling atas yang merupakan penggambaran yang berfungsi untuk memperlihatkan interaksi atas hubungan langsung antara sistem informasi dengan lingkungannya. Diagram konteks menggambarkan sebuah sistem berupa sebuah proses yang berhubungan dengan satu atau beberapa entitas entity.

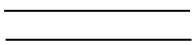
2.2.7.2 *Data Flow Diagram* (DFD)

Data flow diagram (DFD) adalah suatu alat pemodelan yang digunakan untuk memodelkan fungsi dari sistem, menggambarkan secara rinci mengenai sistem sebagai jaringan kerja natar fungsi yang berhubungan satu sama lain dengan menunjukkan dari dan kemana data mengalir serta menyimpannya. Pada umumnya DFD dimulai dari 0,1,2, dst. Level ke-0 disebut dengan diagram konteks yang menggambarkan menggambarkan sistem secara global.

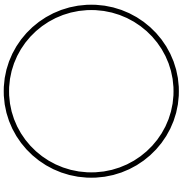
Komponen DFD :



Data Flow / aliran data, diwakili dengan tanda panah. Digunakan untk menunjukkan pergerakan / aliran dari kumpulan data / informasi dari satu bagian ke sistem lainnya .



Data Storage / penyimpanan data, merupakan bagian dari DFD yang digunakan untuk menunjukkan suatu kumpulan dari paket data.



Proses, adalah bagian dari DFD yang mengubah satu atau lebih masukan menjadi keluaran-keluaran

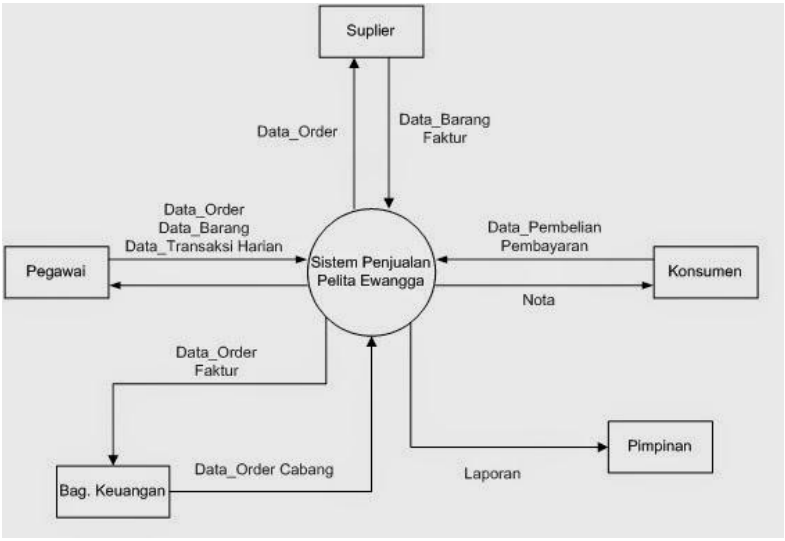


Entitas, adalah seseroang atau sekelompok orang dalam suatu kelompok organisasi atau departemen lain didalam perusahan.

Adapun level dalam DFD adalah :

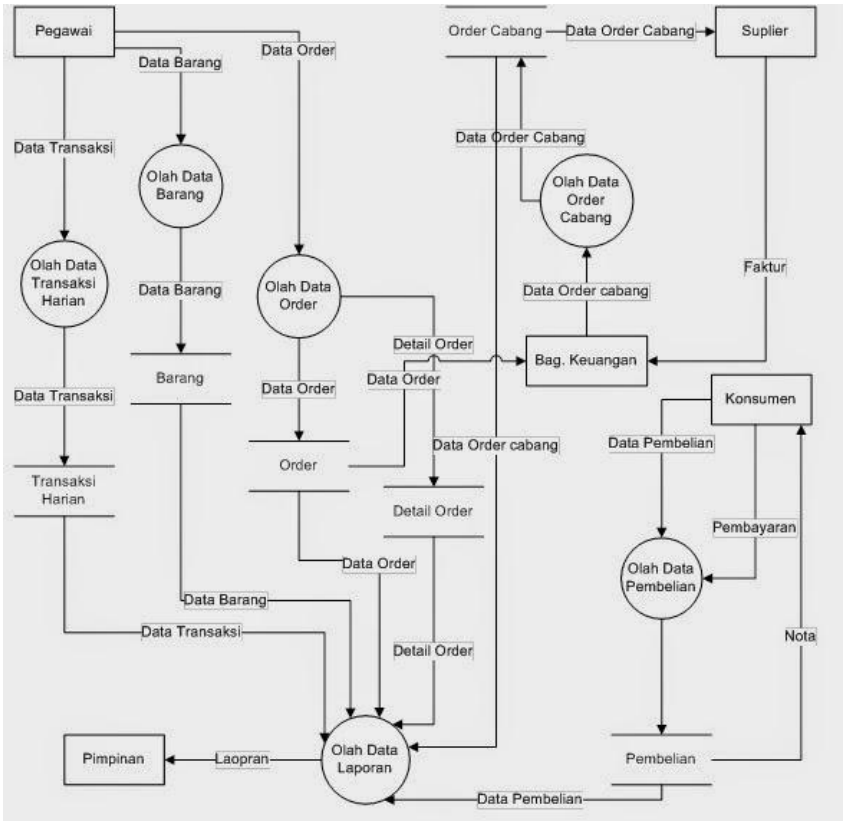
a. Diagram Konteks, merupakan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem.
b. DFD Level 1 merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada didalamnya, dan merupakan pemecahan dari diagram konteks.
c. Diagram Rinci, merupakan diagram yang menguraikan proses apa yang ada dalam diagram DFD Level 1.

Contoh kasus dari penerapan Diagram Konteks :



Gambar 2. 11 Contoh Kasus Penerapan Diagram Konteks

Contoh DFD dari diagram konteks diatas :



Gambar 2. 12 Contoh Kasus Penerapan Diagram Konteks

2.2.7.3 Kamus Data

Kamus data adalah suatu penjelasan tertulis megenai data yang berada dalam *database*. Kamus data ikut berperan untuk menjelaskan arti aliran data dan menyimpan dalam menggambarkan *data flow diagram*, menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut, mendeskripsikan komposisi paket data yang bergerak melalui aliran.

Fungsi kamus data [9] :

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.
2. Mendeskripsikan komposisi paket data (elemen data) yang bergerak melalui aliran data.
3. Mendeskirpsikan komposisi penyimpanan data.
4. Mendeskripsikan nilai dan satuan (struktur data) yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan *detail* antar penyimpanan

Tabel 2. 7 Struktur Kamus Data

Nama Data	Keterangan
Deskripsi	Memilih file yang akan diproses
Bentuk Data	Format file
Struktur Data	Atribut yang digunakan

2.2.7.4 Spesifikasi Proses (*Process Specification)*

Digunakan untuk mendeskripsikan proses yang terjadi pada level paling rendah dari DFD. Struktur spesifikasi proses dapat dilihat di tabel lampiran 2.3. Hubungan dengan *Proses Specification* [11] :

1. Semua proses dalam DFD yang tidak dapat dipecah lagi dan harus didefinisikan dalam *proces specification*.
2. Aliran data masuk (*input*) dan keluar (*output*) dalam DFD dan hubungan ke *data store* harus sesuai dan relevan dalam *proces speci fication*.
3. Setiap elemen data dalam *proces specification* harus :

a. Nama dari aliran data atau *data store*.

b. Atau komponen dalam kamus data atau *data store* yang berhubungan dalam DFD.

Tabel 2. 8 Struktur Spesifikasi Proses

No.Urut	Proses	Keterangan
	No.Proses	
	Nama Proses	
	Source	
	Input	
	Output	
	Destination (tujuan)	
	Logika Proses	