

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1. Profil Perusahaan**

Perusahaan PT. Koloni Semut Pratama bergerak di bidang Jasa pelayanan *Wedding Organizer* yang menyangkut penyewaan gedung, peralatan maupun dekorasi dan makanan atau *catering*. Selain itu disediakan juga jasa fotografer dan video maker dan kebutuhan lainnya sesuai dengan permintaan.

##### **2.1.1 Visi dan Misi**

###### **Visi**

Menjadi perusahaan penyedia layanan *Wedding Organizer* yang dapat diandalkan dan dipercaya.

###### **Misi**

Memberikan layanan terbaik berkenaan dengan jasa penyedia *Wedding Organizer*.

##### **2.1.2 Logo Perusahaan**

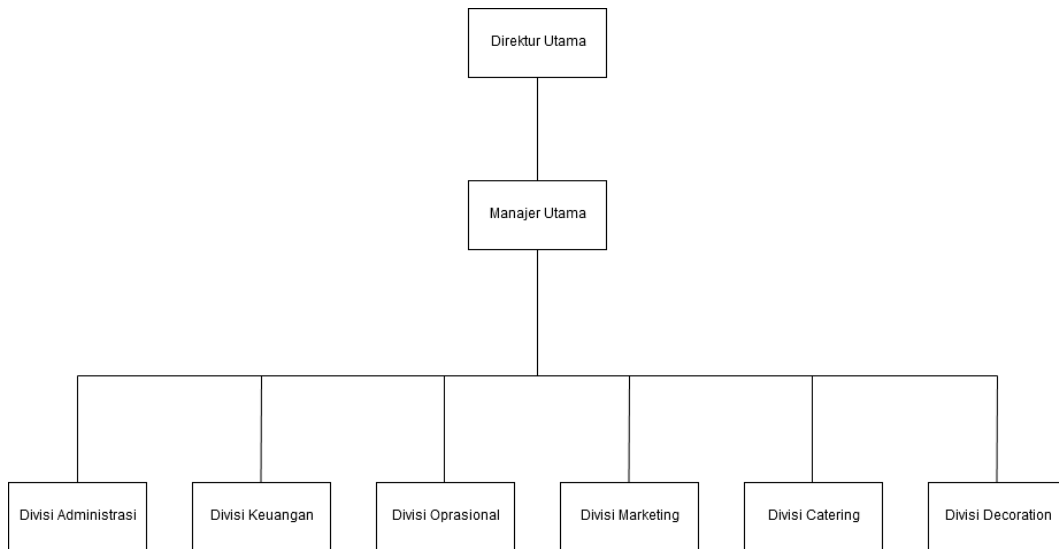
Adapun logo perusahaan PT Koloni Semut Pratama adalah sebagai berikut:



**Gambar 2.1 Logo PT Koloni Semut Pratama**

### 2.1.3 Struktur Perusahaan

Struktur organisasi di perusahaan PT Koloni Semut Pratama sebagai berikut:



**Gambar 2.2 Struktur Perusahaan di PT Koloni Semut Pratama**

## 2.2 Landasan Teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan melakukan penelitian. Landasan teori yang diuraikan merupakan hasil studi literatur yaitu dengan pengumpulan data dengan cara mempelajari jurnal, browsing internet, referensi buku dan bacaan-bacaan lainnya yang berhubungan erat dengan penelitian yang dilakukan.

### 2.2.1 Data

Data adalah fakta mentah mengenai orang, tempat, kejadian, dan hal-hal yang penting dalam organisasi [1].

Data merupakan bahasa, simbol-simbol pengganti lain yang disepakati oleh umum dalam menggambarkan objek, manusia, peristiwa, aktivitas, konsep. Singkatnya data merupakan suatu kenyataan apa adanya [1].

Menurut Laudon & Laudon, data adalah fakta-fakta yang mewakili kejadian-kejadian yang berlangsung dalam organisasi atau lingkungan fisik sebelum didata dan diatur kedalam bentuk yang dapat dipahami dan digunakan orang [2].

## 1. Pengertian Data Menurut Cara Memperolehnya

Jenis data dapat diuraikan berdasarkan cara memperolehnya, yaitu :

### 1. Data *Primer*

Data Primer adalah secara langsung diambil dari objek penelitian oleh peneliti perorang maupun organisasi [4].

Contoh : Mewawancarai langsung penonton bioskop 21 untuk meneliti preferensi konsumen bioskop.

### 2. Data *Sekunder*

Data sekunder adalah data yang didapat tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi yang dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial [4].

## 2. Macam-macam Data Berdasarkan Sumber Data

Macam-macam data dapat dibedakan berdasarkan sumber data yang diperoleh yaitu :

### 1. Data *Internal*

Data *Internal* adalah data yang menggambarkan situasi dan kondisi pada suatu organisasi secara internal [4].

Contoh : Data keuangan, data pegawai, data penjualan suatu perusahaan atau organisasi.

### 2. Data *Eksternal*

Data *eksternal* adalah data yang menggambarkan situasi serta kondisi yang ada di luar organisasi [4].

Contohnya : Data jumlah penggunaan suatu produk pada konsumen tingkat referensi pelanggan, persebaran penduduk dan lain sebagainya.

### 2.2.1 Pengertian Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan saat ini atau saat mendatang [4]. Dapat disimpulkan bahwa :

1. Informasi bermuara pada data.
2. Memberikan suatu nilai tambah atau pengetahuan bagi yang menggunakannya.
3. Dapat digunakan untuk pengambilan keputusan.

### 2.2.2 Konsep Basis Data

Basis data dapat didefinisikan sebagai himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Atau dapat juga didefinisikan sebagai kumpulan data yang saling berhubungan yang disimpan secara bersamaan sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan [6].

*“ Menurut Gordon C. Everest, Database atau basis data merupakan sebuah koleksi atau kumpulan dari data yang bersifat mekanis, terbagi, terdefinisi secara formal dan terkontrol. Pengontrolan dari sistem database tersebut adalah terpusat, yang biasanya dimiliki dan juga dipegang oleh suatu organisasi. “*

*“ Menurut Rogayah, dalam modulnya mengenai sistem basis data, mengatakan bahwa yang dimaksud dengan sistem basis data merupakan suatu sistem yang dapat menyusun dan mengelola record-record menggunakan computer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pengguna untuk proses mengambil keputusan. “*

#### 2.2.2.1 Pengertian Basis Data

Basis Data (database) adalah kumpulan data-data yang umumnya menjabarkan aktivitas-aktivitas dari suatu atau lebih dari satu organisasi yang terkait [5]. Basis Data adalah sebuah cara untuk mendokumentasikan berbagai macam data yang kemudian di manajemen dengan sebuah sistem untuk kemudian disimpan dalam sebuah media penyimpanan. Dalam basis data, data yang ada tidak hanya disimpan begitu saja melainkan akan dikelola dengan sistem pengaturan basis data yang sering disebut *Database Managemenet System (DBMS)*. Dengan begitu pengaksesan data dapat dilakukan dengan mudah dan cepat oleh pengguna.

Keuntungan basis data :

1. Mereduksi redudansi
2. Data dapat di-*share* antar aplikasi.
3. Standarisasi dapat dilakukan.
4. Batasan *security* dapat diterapkan.
5. Mengelola integritas data.

## 6. Independensi data.

### 2.2.2.2 Bahasa dalam Basis Data

Ada Beberapa Bahasa yang digunakan dalam basis data diantaranya adalah sebagai berikut :

#### 1. *Data Definiton Language* (DDL)

*Data Definition Language* (DDL) ini adalah sekumpulan perintah-perintah yang terdapat dalam SQL yang berfungsi untuk menspesifikasikan atau menggambarkan suatu skema basis data. Dengan Bahasa ini hal-hal yang dapat dilakukan adalah membuat table baru, membuat indeks, mengubah struktur table, menentukan struktur penyimpanan table, dan lain-lain. Hal yang dasar yang dapat dilakukan DDL ini adalah menciptakan, mengubah, dan menghapus basis data.

#### 2. *Data Manipulation Language* (DML)

*Data Manipulation Language* (DML) adalah sekumpulan perintah-perintah dapat dikatakan seperti Bahasa pemrograman atau sintak yang dapat digunakan untuk mengubah, memanipulasi, dan mengambil data pada basis data. Dapat melakukan aksi crud seperti menghapus, mengubah, menyisipkan dan mengambil data. DML dibagi menjadi 2 jenis yaitu :

##### A. Prosedural

Prosedural menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkan dari data tersebut.

##### B. Non-Prosedural

Non-prosedural menuntut pengguna menentukan terlebih dahulu data apa saja yang diperlukan dan yang dibutuhkan, tetapi tidak perlu mendeskripsikan atau menyebutkan bagaimana cara data tersebut mendapatkannya.

##### C. *Transaction Control*

*Transaction Control* adalah bahasa pada basis data yang dapat mengatur transaksi dimana transaksi dilakukan oleh *Data Manipulation Language* (DML). *Transaction Control* ini memiliki peran yang sangat besar untuk menentukan suatu aktivitas dilakukan atau tidaknya perubahan-perubahan

data yang terdapat dalam basis data. Contoh dari *transaction control* adalah perintah *commit* dan *rollback*.

### 2.2.2.3 DBMS (Database Management System)

DBMS (*Database Management System*) atau disebut dengan sistem manajemen basis data, berfungsi sebagai suatu sistem dalam pengolahan basis data sehingga menjadikan sebuah data menjadi sebuah informasi baru yang berguna. Sistem ini memungkinkan untuk menyusun, mengolah dan memperbaharui data yang terdapat dalam suatu basis data. DBMS memiliki kemampuan untuk mengolah data dalam jumlah yang besar dan memanipulasi data dengan cepat dan mudah. Tujuan utama dari DBMS adalah menyediakan cara untuk menyimpan dan mengambil informasi dari *database* dengan baik, nyaman, dan efisien [5].

### 2.2.2.4 RDBMS (Relational Database Management System)

RDBMS (*Relational Database Management System*) merupakan sekumpulan data yang saling berhubungan yang disimpan sedemikian rupa sehingga mudah diambil informasinya oleh pengguna.

Ada Tiga prinsip dalam RDBMS :

#### 1. *Data Definition*

Mendefinisikan data-data yang akan dibuat, seperti nama table dan *field* serta keterangan dari *field* tersebut.

#### 2. *Data Manipulation*

Melakukan manipulasi data yang sudah dibuat, seperti mengubah na a *field* dan keterangannya, dan meghapus *record*.

#### 3. *Data Control*

Bagian ini berkenaan pada mengendalikan data kepada siapa saja yang bisa melihat isi data.

### 2.2.2.5 Database Administrator

*Database Administrator* adalah orang memiliki kemampuan dalam melakukan *control* utama terhadap keseluruhan sistem basis data (mencakup data dan program) yang mempunyai fungsi sebagai berikut :

1. Pendefinisian skema.
2. Pendefinisian struktur penyimpanan dan metode akses.
3. Modifikasi skema dan organisasi fisik.

4. Pemberian otorisasi bagi pengaksesan data
5. Mendefinisikan bagian basis data yang mana dapat diakses oleh seorang
6. Pengguna, termasuk operasi-operasu yang dapat dilakukan.
7. Spesifikasi batasan integrasi.

### **2.2.2 Database**

*Database* atau Basis data merupakan kumpulan data yang disimpan secara sistematis didalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit [1].

Basis data (Database) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti [1]:

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 (redun-dansasisi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file, table, arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

### **2.2.3 Database Management System (DBMS)**

Aplikasi yang digunakan untuk membangun sebuah sistem basis data yang memungkinkan Pengguna untuk mendefinisikan, membuat, memelihara dan mengontrol akses ke database merupakan pengertian dari *Database Management System* (DBMS). DBMS membantu dalam pemeliharaan dan pengolahan kumpulan data dalam jumlah besar, sehingga dengan bantuan DBMS tidak akan menimbulkan kesalahan dan dapat digunakan oleh pengguna sesuai dengan kebutuhan. Bahasa

yang digunakan dalam *database* biasanya terdiri atas perintah-perintah yang diformulasikan dalam bentuk *query* sehingga perintah tersebut akan diproses oleh DBMS [1].

#### 2.2.4 Data Warehouse

Pengertian *Data Warehouse* dapat bermacam-macam namun mempunyai inti yang sama, seperti pendapat beberapa ahli berikut ini :

*Data warehouse* adalah koleksi data yang mempunyai sifat berorientasi subjek, terintegrasi, *time-variant*, dan bersifat tetap dari koleksi data dalam mendukung proses pengambilan keputusan *management* [3].

*Data warehouse* merupakan *database* relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, biasanya mengandung histori data dari proses transaksi dan bisa juga data dari sumber lainnya. *Data warehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung/konsolidasi data dari berbagai macam sumber [3].

*Data warehouse* merupakan metode dalam perancangan database, yang menunjang DSS (*Decision Support System*) dan EIS (*Executive Information System*). Secara fisik *data warehouse* adalah database, tapi perancangan *data warehouse* dan database sangat berbeda. Dalam perancangan database tradisional menggunakan normalisasi, sedangkan pada *data warehouse* normalisasi bukanlah cara yang terbaik [3].

Dari definisi-definisi yang dijelaskan tadi, dapat disimpulkan *data warehouse* adalah database yang saling bereaksi yang dapat digunakan untuk *query* dan analisis, bersifat orientasi subjek, terintegrasi, *time-variant*, tidak berubah yang digunakan untuk membantu para pengambil keputusan.

##### 2.2.4.1 Tujuan Data Warehouse

Tujuan dari *data warehouse* adalah sebagai berikut [9] :

1. Memberikan kemudahan untuk mengakses informasi yang ada.

Kemudahan disini berbicara tentang efisiensi. *Data warehouse* harus efisien sehingga dengan mudah dipahami oleh pengguna bukan hanya *olive developer* saja. Selain itu, pengguna juga dapat mengkombinasikan data dalam *data warehouse* dengan berbagai cara (*slicing and dicing*). Untuk mengakses data



*warehouse* disarankan sebaiknya dapat dilakukan dengan sederhana dan mudah dioperasikan.

2. Menyediakan informasi yang konsisten.

Data *warehouse* hanya berisi informasi-informasi yang relevan bagi kebutuhan pengguna untuk mengambil suatu keputusan. Oleh karena itu, kredibilitas data yang terdapat dalam data *warehouse* harus dapat dipertanggungjawabkan.

3. Mampu beradaptasi dan tahan terhadap perubahan.

Perubahan-perubahan yang terjadi harus dapat diatasi oleh data *warehouse*. Dengan kata lain, data *warehouse* harus dirancang agar mampu menghadapi setiap perubahan dengan terencana. Hal ini berarti perubahan yang terjadi tidak boleh merusak atau mengganggu data dan aplikasi yang telah ada sebelumnya.

4. Mampu mengamankan informasi.

Informasi yang tersimpan dalam data *warehouse* harus tersimpan dengan aman. Dengan kata lain, informasi tersebut tidak boleh sampai jatuh ke tangan yang salah. Oleh karena itu, data *warehouse* harus mampu mengendalikan setiap akses dari informasi yang ada.

5. Mampu memberikan dukungan dalam pengambilan suatu keputusan.

Ini merupakan tujuan yang paling penting dan harus ada dalam setiap pembuatan data *warehouse*. Data *warehouse* bisa digambarkan sebagai kumpulan teknologi pendukung suatu keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan sebuah informasi dapat membuat keputusan dengan cepat dan tepat.

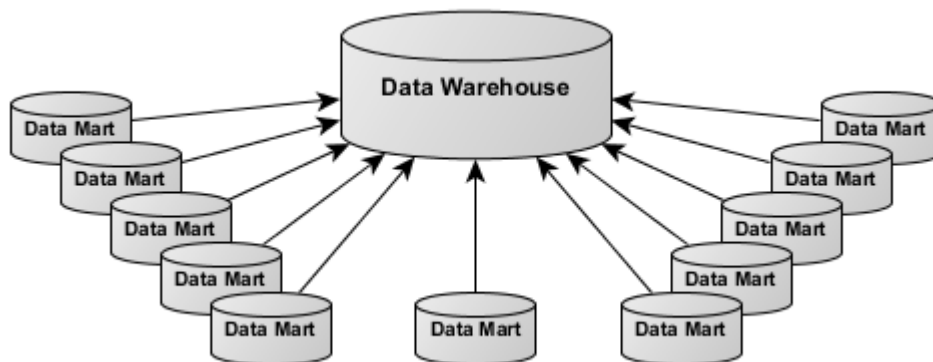
#### 6. *User friendly.*

Seperti pada tujuan data *warehouse* pertama, data *warehouse* harus dirancang agar dapat dioperasikan dengan mudah oleh pengguna. Tidak seperti sistem operasional dimana seringkali pengguna tidak memiliki pilihan yang lain kecuali menggunakan sistem baru, akan terjadi pengguna data *warehouse* biasanya merupakan pilihan. Oleh karena itu, proses penentuan pengguna data *warehouse* merupakan factor yang sangat penting.

#### 2.2.5 *Data Mart*

*Data Mart* merupakan unit bagian dari *Data Warehouse* secara keseluruhan yang berada pada *layer* akses berorientasikan kepada spesifik proses bisnis dan kebutuhan dari unit organisasi tempat dimana *Data Warehouse* tersebut diimplementasikan.

Terdapat hubungan antara *Data Mart* dengan *Data Warehouse*. *Data Mart* merupakan bagian dari *Data Warehouse*, untuk mewujudkan hubungan ini dilibatkan adanya proses menggunakan OLAP, ELT, analisis, *Query*, *Reporting*, dan *Data Mining*.



**Gambar 2.3 Data Warehouse dan Data Mart**

##### 2.2.5.1 *Mafaat Data Mart*

*Data mart* diperlukan bagi para pengambil keputusan manajemen dari suatu organisasi/perusahaan untuk sebuah bidang divisi tertentu. Dengan adanya *data mart*, akan mempermudah pembuatan aplikasi-aplikasi DSS (*Decision Support System*) dan EIS (*Executive Information System*) karena kegunaan dari *data mart* adalah khusus untuk membuat suatu *database* yang dapat digunakan untuk

mendukung proses analisa (OLAP), mengambil keputusan, pembuatan laporan, penggalian informasi baru (*Data Mining*) dari banyak data dan proses *executive* informasi [3].

Beberapa kegunaan data *mart*:

a. Pembuatan laporan

Pembuatan laporan merupakan salah satu kegunaan *data mart* yang paling umum dilakukan. Dengan menggunakan query sederhana didapatkan laporan perhari,perbulan, pertahun atau jangka waktu kapanpun yang diinginkan.

b. *On-Line Analytical Processing* (OLAP)

Dengan adanya *data mart*, semua informasi baik detail maupun hasil summary yang dibutuhkan dalam proses analisa mudah didapat. OLAP mendayagunakan konsep data multidimensi dan memungkinkan para pengguna menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL. Hal ini dimungkinkan karena pada konsep multidimensi, maka data yang berupa fakta yang sama bisa dilihat dengan menggunakan fungsi yang berbeda. Fasilitas lain yang ada pada software OLAP adalah fasilitas *rool-up* dan *drill-down*. *Drill-down* adalah kemampuan untuk melihat detail dari suatu informasi dan *roll-up* adalah kebalikannya.

c. *Data mining*

*Data mining* merupakan proses untuk menggali (*mining*) pengetahuan dan informasi baru dari data yang berjumlah banyak pada *data mart*, dengan menggunakan kecerdasan buatan (*Artificial Intelegence*), statistik dan matematika. *Data mining* merupakan teknologi yang diharapkan dapat menjembatani komunikasi antara data dan pemakainya.

Beberapa solusi yang diberikan data mining antara lain :

1. Menebak target pasar

*Data mining* dapat mengelompokkan (*clustering*) model-model pembeli dan melakukan klasifikasi terhadap setiap pembeli dan melakukan klasifikasi terhadap setiap pemebeli sesuai dengan karakteristik yang diinginkan.

2. Melihat pola beli dari waktu ke waktu

*Data mining* dapat digunakan untuk melihat pola beli dari waktu ke waktu.

3. *Cross-market analysis*

*Data mining* dapat dimanfaatkan untuk melihat hubungan antara satu produk dengan produk lainnya.

4. Profil pelanggan

*Data mining* bisa membantu pengguna untuk melihat profil pembeli sehingga dapat diketahui kelompok pembeli tertentu cenderung kepada suatu produk apa saja.

5. Informasi *summary*

*Data mining* dapat membuat laporan *summary* yang bersifat multidimensi dan dilengkapi dengan informasi statistik lainnya.

6. Proses informasi *executive*

*Data mart* dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan *data mart* segala laporan telah diringkaskan dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan. Informasi dan data pada laporan *data mart* menjadi target *informative* bagi user.

#### 2.2.5.2 Konsep Dasar *Data Mart*

*Data mart* adalah kumpulan macam-macam data yang *subject oriented*, *integrated*, *time variant*, dan *nonvolatile* dalam mendukung proses pembuatan keputusan [4].

*Data mart* sering diintegrasikan dengan berbagai sistem aplikasi untuk mendukung proses laporan dan analisis data dengan menyediakan data historis, yang menyediakan infrastruktur bagi EIS dan DSS. *Data mart* memiliki konsep dasar yang sama dengan *data mart* yaitu sebagai berikut:

a. *Subject Oriented*

*Data mart* diorganisasikan pada subjek-subjek utama, seperti pelanggan, barang, dan penjualan. Berfokus pada model dan analisis pada data untuk membuat keputusan, jadi bukan pada setiap proses transaksi atau bukan pada OLTP. Menghindari data yang tidak berguna dalam mengambil suatu keputusan.

b. *Integrated*

Dibangun dengan menggabungkan atau menyatukan data yang berbeda. relational *database*, *flat file*, dan *on-line transaction record*. Menjamin konsistensi dalam penamaan, struktur pengkodean, dan struktur atribut diantara data satu sama lain.

c. *Time-variant*

Data disimpan untuk menyediakan informasi dari perspektif historical, data yang tahun - tahun lalu atau 4 - 5 tahun. Waktu adalah elemen kunci dari suatu data *mart*.

d. *Non Volatile*

Setiap kali proses perubahan, data akan di tampung dalam tiap-tiap waktu. Jadi tidak di perbaharui terus menerus. Data *mart* tidak memerlukan pemrosesan transaksi dan *recovery*. Hanya ada dua operasi *initial loading of data* dan *access of data*.

e. *Accessible*

*Accessible* secara langsung berarti mudah diakses, disini menyatakan *Data Warehouse* beserta data-data di dalamnya harus dapat diakses dengan mudah oleh pengguna.

### 2.2.5.3 Proses Data Mart

*Data Warehouse* dibangun dengan mengintegrasikan data-data yang berasal dari berbagai sumber data, yaitu database operasional. Dalam suatu perusahaan, data operasional biasanya berada pada daerah kekuasaan departemen masing-masing dalam bentuk database OLTP. Untuk melakukan proses integrasi ini data warehouse menggunakan suatu aplikasi yang disebut ETL (*extract, Transform, Load*) [5].

1. *Extract*.

Pada tahap ini *Extraction*, dilakukan ekstrasi data yang berasal dari berbagai sumber data, sumber-sumber data tersebut berasal dari berbagai *database* yang menggunakan sistem OLAP, website, aplikasi, flat file dan lainnya, pada tahap *Extraction* ini bertujuan untuk mengidentifikasi bagian-bagian data yang tepat, yang akan dimasukkan kedalam *Work Flow* dari ETL untuk pemrosesan selanjutnya.

## 2. *Tranform.*

Jika pada tahap *Extrction* semua data yang dibutuhkan berhasil dikumpulkan dari berbagai sumber data, namun belum memiliki keseragaman format data, maka pada tahap kedua ini penyamaan format data tersebut dilakukan melalui tahapan proses yang disebut *Tranformation*.

Terdapat tiga buah skema atau masalah yang ada pada saat proses penyeragaman format data diantaranya:

### a. *Schema Level Problem.*

Pada level ini, dilakukan penyelesaian konfil-konflik penamaan dalam data dimana kemungkinan terjadi nama yang sama dalam subjek yang berbeda.

### b. *Record Level Problem.*

Pada level *Record Level Problem*, permasalahan terletak pada duplikasi sebuah data (*Redudancy*) pada Record data yang berasal dari berbagai sumber data tersebut.

### c. *Value Lebel Problem.*

Masalah yang terdapat pada level ini adalah masalah nilai (*Value*) yang berbeda beda dari setiap data, sebab dari format data dan sumber data tersebut berbeda-beda. Misal di dalam sumber data A format penulisan jenis kelamin adalah Laki-laki dan Perempuan, sedangkan di dalam sumber data B format penulisan jenis kelamin adalah L dan P. Hal lainnya biasanya pada format untuk tanggal atau *date*.

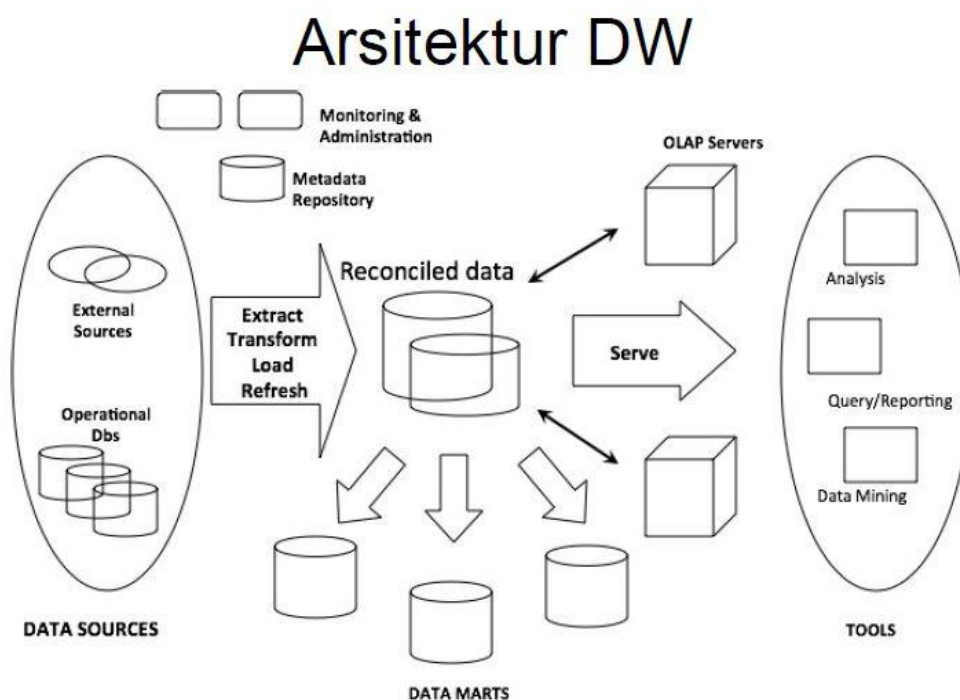
## 3. *Load.*

Pada tahap terakhir ini, data yang sudah diseragamkan formatnya akan diteruskan ke tahap muka aplikasi. Pada tahap ini sistem yang ada pada ETL melakukan tiga hal, *Load Up Data* untuk menyalurkan data-data yang telah diintegrasikan dan diseragamkan tersebut ke dalam tampilan muka aplikasi. *Load Insert Data* untuk memasukan data yang diteruskan tersebut kedalam *database* pada *Data Warehouse*. *Load Bulk Data* untuk meneruskan *Bulk Data* yang memuat data-data dan informasi di dalamnya.

#### 2.2.5.4 Arsitektur Data Warehouse Dan Data Mart

Arsitektur *Data Warehouse* didefinisikan pertama kali oleh Ken Orr di dalam paper publikasinya tahun 1999 berjudul *Data Warehouse Technology*, Ken Orr berpendapat bahwa Arsitektur *Data Warehouse* merupakan sebuah cara yang dilakukan oleh arsitek *Data Warehouse* di dalam sebuah organisasi untuk merepresentasikannya didalam bentuk bagan, gambar atau desain yang memuat segala hal mengenai struktur data, komunikasi, pemrosesan dan presentasi dari komputasi antar komputer di dalam sistem *Data Warehouse*.

Tujuan dari arsitektur *Data Warehouse* adalah untuk memudahkan pengembangan pemeliharaan sistem oleh bagian IT, serta memudahkan pengguna di dalam memahami dan menggunakan sistem. *Data Warehouse* juga berkaitan dengan strategi yang berkaitan dengan strategi di dalam pengembangan *Data Warehouse* yang meliputi desain, pengembangan, pengolahan dan pemeliharaan sistem [6].



**Gambar 2.4 Arsitektur Data Warehouse dan Data Mart**

1. *Source Data*  
Merupakan sumber data yang akan diolah, baik itu data eksternal seperti dokumen tertulis ataupun data operasional seperti *database*.
2. Proses ETL

Merupakan sebuah proses yang diperlukan sebelum data *source* masuk ke dalam *Data Warehouse*, dimana prosesnya terdiri dari *Extract*, *Transform* dan *load*.

3. *Data Warehouse*

Merupakan tempat penyimpanan data yang multidimensional, dimana data yang tersimpan berupa *metadata*, *summary* data dan *row* data.

4. *Data Mart*

Merupakan bagian dari sebuah data *warehouse*. Seluruh data *mart* jika digabungkan akan menjadi satu data *warehouse*.

5. *Tool*

Merupakan sebuah proses akhir dalam Arsitektur *Data Warehouse*, di sini *Data Warehouse* bisa dimanfaatkan dalam berbagai cara seperti menggunakannya dalam proses analisis, membuat sebuah laporan atau bahkan proses data mining.

### 2.2.6 Multi Dimensional Modeling

*Multi dimensional modeling* adalah teknik untuk memvisualisasi model data sebagai suatu kumpulan dari ukuran yang dideskripsikan dengan aspek-aspek bisnis. Hal ini khususnya sangat berguna untuk meringkas dan menyusun data dan memperlihatkan data untuk mendukung para analis data. *Dimensional modeling* mempunyai beberapa konsep.

1. *Fact*

*Fact* adalah suatu koleksi dari relasi data-data items, terdiri dari ukuran-ukuran dan konteks data. Setiap *fact* biasanya merepresentasikan sebuah bisnis item, suatu transaksi bisnis, atau sebuah kejadian yang dapat digunakan dalam analisis bisnis atau proses bisnis.

2. *Dimensions*

*Dimensions* adalah suatu koleksi dari anggota atau unit-unit data dengan tipe yang sama. Dalam sebuah diagram, suatu dimensi biasanya direpresentasikan dengan suatu axis. Dalam dimensional model, semua data menunjukkan *fact table* yang diasosiasikan dengan satu dan hanya satu member dari setiap multiple dimensions. Jadi dimensi menunjukkan latar belakang kontekstual dari *fact*. Banyak proses analisis yang digunakan untuk menghitung (*quality*) dampak dari dimensi pada *fact*.



### 3. *Measures*

Suatu *measures* (ukuran) adalah suatu besaran (angka numerik) atribut dari sebuah *fact*, yang menunjukkan *performance* atau *behavior* (tingkah laku) dari bisnis secara relatif pada suatu dimensi. Angka atau nomor yang ditunjukkan disebut dengan *variabel*. Sebagai contoh ukuran dari penjualan dalam bentuk uang, besarnya penjualan, jumlah pengadaan, banyaknya transaksi dan lainnya. Suatu ukuran dijelaskan dengan kombinasi dari member dari suatu dimensi dan diletakkan dalam *fact*.

#### 2.2.7 Online Analytical Processing (OLAP)

*Online Analytical Processing* (OLAP) merupakan salah satu tools yang digunakan untuk mengakses informasi dalam *Data Warehouse*. Teknologi OLAP memungkinkan *Data Warehouse* digunakan secara aktif untuk proses online analisis, memberikan respon yang cepat terhadap analytical queries yang kompleks.

Data model multidimensional dan teknik agregasi data yang dimiliki oleh OLAP dapat mengatur dan membuat kesimpulan dari data dalam jumlah besar sehingga dapat dievaluasi secara cepat dengan menggunakan online analysis dan graphical tool. Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk mendukung proses analisis secara real time.

##### 2.2.7.1 OLAP Services

OLAP *services* adalah suatu yang menggambarkan beberapa perangkat yang bekerja bersama-sama untuk membantu dalam menganalisa data yang ada. Informasi dari data warehouse diekstrak secara periodik dan digunakan untuk mengupdate objek yang ada dalam OLAP *service*. OLAP *server* mengambil data dari data warehouse dan relational OLAP serta mengupdate informasi yang disimpan dalam multidimensional OLAP (MOLAP). Setelah informasi dikumpulkan maka pivot table service yang bekerja sama dengan *excel pivot table* atau perangkat lain yang mendukung OLE DB dengan ekstensi OLAP dapat melakukan akses dan memanipulasi data yang ada.

##### 2.2.7.2 Teknik Penyimpanan Data dalam OLAP

OLAP digunakan untuk memproses informasi dan menampilkannya dalam bentuk multidimensi. Walaupun data yang ditampilkan dalam satu bentuk

namun tidak berarti data-data yang ada disimpan dalam satu bentuk pula. Ada tiga teknik dasar untuk menyimpan data OLAP:

1. *Multidimensional* OLAP (MOALP)

Salah satu cara umum yang digunakan untuk menyimpan data adalah dalam basis data multidimensional. Tidak seperti basis data relasional yang menyimpan informasi dalam sejumlah baris dalam tabel. Sebuah basis data multidimensional menyimpan informasi dalam sejumlah array multidimensional. Karena dimensi dapat diakses secara mudah, maka user dapat melakukan *query* pada basis data MOLAP dengan sangat cepat. Selain mengandung data mentah, basis data MOALP juga mengandung agregasi data sehingga dapat memberikan respon yang cepat terhadap *query*.

2. *Relational* OLAP (ROLAP)

Partisi menggunakan tabel relasional dalam data warehouse untuk menyimpan agregasi, sedangkan detail dari fact table tersimpan dalam data warehouse *fact table*. Banyak orang berpendapat bahwa basis data yang dirancang secara khusus untuk sebuah keperluan analisis tidak dibutuhkan karena sebuah basis data relasional sudah cukup mampu untuk menampilkan informasi OLAP. Hal ini hanya berlaku pada tingkat tertentu saja, pada sebuah basis data yang terdiri dari ribuan atau ratusan ribu *records* maka menampilkan informasi OLAP akan menjadi sebuah masalah karena banyak data yang harus dihapus di *wuery*. Dan hal inilah yang menjadi keterbatasan partisi ROLAP.

3. *Hybryd* OLAP (HOLAP)

Partisi menggunakan struktur MOLAP untuk menyimpan agregasi dan meninggalkan detail dari *fact table* dalam partisi ROLAP. OLAP *service* dapat menggunakan data MOLAP dan ROLAP secara simultan untuk memecahkan suatu *query*.

### 2.2.7.3 OLAP Cube (Kubus OLAP)

Objek utama yang tersimpan dalam sebuah basis data OLAP adalah *cube* (kubus). Sebuah kubus merupakan representasi multidimensi dari sekumpulan data, yang mengandung data secara detail maupun rangkumannya. Sebuah basis data OLAP dapat memiliki beberapa buah kubus sesuai dengan yang dibutuhkan, yang menggambarkan data yang ada dalam data warehouse.

Sebuah kubus dibangun menggunakan dua komponen utama yaitu ukuran (*measures*) dan dimensi. Ukuran merupakan nilai numerik dari *fact table* dalam *Data Warehouse* seperti harga dari unit maupun kualitas dari item. Sedangkan dimensi menggambarkan kategori dari ukuran yang ada. Seperti bagaimana ukuran berubah setiap waktu. Beberapa operasi yang dapat dilakukan pada kubus atau reprot multidimensi ini adalah:

1. *Processing*

Operasi *processing* pada kubus antara lain melakukan loading dan refreshing data pada kubus yang dilakukan oleh OLAP *service engine*. Tabel dimensi dibaca pertama kali untuk mengumpulkan level dengan anggota dari data aktual, setelah itu dilanjutkan dengan pembacaan *fact table* dan diikuti dengan menghitung spesifik agregasi, dan hasil yang dapat disimpan dalam kubus untuk diproses oleh user.

2. *Slice and dice*

*Slice and dice* memungkinkan untuk melihat kubus dari sudut pandang yang berbeda. Dengan *slice and dice* dapat ditentukan dimensi apa yang hendak ditampilkan dan bagaimana mereka ditampilkan. Hal inilah yang menjadi keunggulan OLAP. Dengan melihat kubus dari sudut pandang yang berbeda maka akan dapat dipelajari banyak hal dari data yang dimiliki.

3. *Drill down*

Sebagian besar informasi yang ditampilkan dalam OLAP merepresentasikan kesimpulan yang lebih detail. Drilling down merupakan teknik untuk memecahkan sebuah informasi menjadi beberapa informasi yang lebih detail. Sebagai contoh, jika dilakukan drill down terhadap data tahunan maka akan dapat diperoleh data dalam catur wulan dan tri wulan.

4. *Consolidation (Roll Up)*

*Consolidation* atau lebih dikenal dengan *roll up* merupakan kebalikan dari *drill down*. Data-data sebelumnya dilihat dalam format triwulan akan dapat dilihat dalam format tahunan. Dengan *roll up* data dapat dilihat secara lebih global.

5. *Pivoting*

*Pivoting* merupakan suatu teknik untuk saling menukarkan dimensi data. Dengan melakukan *pivoting*, maka dapat diamati suatu informasi atau data dari

sudut pandang yang berbeda, sehingga diharapkan akan dapat memperjelas analisis yang dilakukan.

#### 6. *Filtering*

*Filtering* merupakan suatu teknik untuk menyaring informasi yang dibutuhkan, sehingga para analisis tidak dibingungkan oleh banyaknya informasi yang tersedia.

### 2.2.8 Microsoft Visual Studio

*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 *console*, aplikasi windows, ataupun aplikasi web. Visual studio mencakup kompiler, dan dokumentasi. Kompiler yang dimasukkan ke dalam paket visual studio antara lain visual C++, Visual C#, Visual basic.

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 *NET framework*). Selain itu, visual studio juga dapat digunakan untuk mengembangkan aplikasi silverlight, aplikasi windows mobile.

### 2.2.9 SQL Server

Microsoft *SQL server* adalah sebuah sistem manajemen basis data relasional (RDBMS) produk microsoft. Bahasa kueri utamanya adalah transact-sql yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh microsoft. Umumnya *SQL server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL server* pada basis data besar.

Microsoft *SQL server* dan *sybase/ASE* dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*tabular Data Stream*). Selain dari itu, *SQL server* juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai driver JDBC untuk bahasa pemrograman java. Fitur yang lain dari *SQL server* ini adalah kemampuannya untuk membuat basis data mirroring dan clustering. *SQL Server* adalah sistem manajemen database relasional (RDBMS) yang dirancang

untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada level yang sangat umum, sebuah *client* adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. Sedangkan sebuah *server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

Pada dasarnya pengertian dari *SQL Server* itu sendiri adalah bahasa yang dipergunakan untuk mengakses data dalam basis data relation. Bahasa ini secara *defacto* adalah bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini dalam manajemen datanya. *SQL server 2005* merupakan salah satu produk dari *Relational Database Management System (RDBMS)*.

*SQL Server* terdiri atas beberapa komponen sebagai berikut:

- a. *Relational Database Engine* : komponen utama atau jantung *SQL Server*.
- b. *Analysis Services* : Basis dari solusi intelijen bisnis yang ampuh (*powerful*), dan mendukung aplikasi-aplikasi OLAP (*online analytical processing*), serta data mining.
- c. *Data Transformation Service (DTS)*: sebuah mesin untuk membuat solusi ekspor dan impor data, serta untuk mentransformasi data ketika data tersebut ditransfer.
- d. *Notification Services*: sebuah framework untuk solusi dimana pelanggan akan dikirim notifikasi ketika sebuah event muncul.
- e. *Reporting Services*: service yang akan mengambil data dari *SQL Server*, dan menghasilkan laporan-laporan.
- f. *Service broker*: sebuah mekanisme antrian yang akan menangani komunikasi berbasis pesan diantara service.
- g. *Native HTTP Support*: dukungan yang memungkinkan *SQL server* yang (jika diinstall pada *Windows Server 2003*) akan merespon request terhadap *HTTP endpoint*, sehingga memungkinkan pembangunan sebuah web service untuk *SQL Server* tanpa menggunakan *IIS*.

- h. *SQL server Agent* : akan mengotomatiskan perawatan database dan mengatur task, event dan alert.
- i. *NET CLR (Common Language Runtime)*: akan memungkinkan pembuatan solusi menggunakan *managed code* yang ditulis dalam salah satu bahasa .NET.
- j. *Replication*: serangkaian teknologi untuk menjalin dan mendistribusikan data dan objek database dari sebuah database ke database lain, dan melakukan sinkronisasi untuk menjaga konsistensinya.
- k. *Full-Text Search*: memungkinkan pengindeksan yang cepat dan flexibel untuk query berbasis kata kunci (terhadap data teks yang disimpan dalam database).

### 2.2.10 Diagram Konteks

Diagram konteks adalah level teratas dari diagram aliran data, yaitu diagram yang tidak detail dari sebuah sistem informasi yang menggunakan aliran-aliran data kedalam dan keluar entitas eksternal. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada di luar sebuah sistem yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan dan keluaran sistem. Diagram konteks ini direpresentasikan dengan lingkaran tunggal yang mewakili seluruh sistem.

### 2.2.11 Data Flow Diagram

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

#### 1. Komponen Terminator

*Terminator* mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya *terminator* dikenal dengan nama entitas luar (*external entity*). Terdapat dua jenis *terminator*:

- A. *Terminator Sumber (source)* merupakan terminator yang dijadikan sebagai sumber.

B. *Terminator* Tujuan (*sink*) merupakan terminator yang menjadi tujuan dari data-data yang dibutuhkan.

*Terminator* dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. *Terminator* dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan. Ada tiga hal penting yang harus diingat tentang *terminator* :

- A. *Terminator* merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan *terminator* dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
- B. Profesional sistem tidak dapat mengubah isi atau cara kerja organisasi, atau prosedur yang berkaitan dengan *terminator*.
- C. Hubungan yang ada antar *terminator* yang satu dengan yang lain tidak digambarkan pada DFD.

## 2. **Komponen Proses**

Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output. Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan objek). Ada beberapa hal yang perlu diperhatikan tentang proses:

- A. Proses harus memiliki input dan output.
- B. Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
- C. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

Umumnya kesalahan proses di DFD adalah :

- A. Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan black hole (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam.

- B. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan *miracle* (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input.

### 2.2.12 Kamus Data

Kamus data merupakan kumpulan data-data. Kamus data adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Kamus data merupakan tempat penyimpanan definisi dari aliran-aliran data, file-file dan proses-proses dalam sebuah sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat memberikan informasi mengenai definisi struktur pemakaian masing-masing elemen, dapat mendefinisikan data yang mengalir di sistem dengan lengkap, dapat menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen.

Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan alokasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Kamus data mendefinisikan data elemen dengan cara:

1. Menguraikan arti dari alur data dan data *store* dalam DFD .
2. Menguraikan komposisi paket data pada alur data ke dalam alur yang lebih *elementary* (kecil). Contoh : alamat langganan yang terdiri dari nama jalan, kota dan kode pos.
3. Menguraikan komposisi paket data dalam data *store*.
4. Menspesifikasikan nilai dan unit informasi dalam alur data dan data *store*.
5. Menguraikan hubungan yang terinci antara data *store* dalam suatu *entity relationship diagram* (ERD).