BAB 2
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Optik Yuda adalah sebuah usaha yang bergerak dalam bidang pelayanan kesehatan, khususnya kesehatan mata. Optik Yuda bergerak sebagai pusat layanan kacamata yang bertujuan untuk memenuhi kebutuhan para konsumen yang memerlukan alat bantu penglihatan yakni kacamata. Optik Yuda melayani pasien umum maupun pasien peserta BPJS.

2.1.1 Sejarah Perusahaan

Optik Yuda didirikan oleh Ade Wahyudin pada tahun 1983. Sebelum mendirikan Optik Yuda, Ade Wahyudin telah berkecimpung dalam bidang optik dengan ikut bekerja dengan orang lain di optik Jalan Melawai, Jakarta. Berbekal pengalaman itu lah Ade Wahyudin mendirikan usaha sendiri, yang dahulu dinamakan Optik Joeda lalu berubah nama menjadi Optik Yuda seiring dengan berkembangnya zaman. Pada awalnya, optik Yuda dibuka di rumah. Seiring berjalannya waktu, mulai membuka toko sendiri di jalan Sukajadi. Lalu membuka cabang di jalan Kopo dan Jalan Dalem Kaum. Namun, pada tahun 1997-1998 dikarenakan krisis moneter yang melanda Indonesia maka toko di Jalan Dalem Kaum terpaksa harus ditutup. Selang beberapa tahun kemudian, mencoba peruntungan lagi untuk membuka optik di Jalan Buah Batu, namun itupun tidak berlangsung lama dan akhirnya harus tutup kembali. Situasi ekonomi yang tidak menentu membuat dunia usaha pun ikut merasakan dampaknya sehingga toko yang beralamat di jalan Kopo pun harus ditutp. Hingga pada tahun 2004, Ade Wahyudin memutuskan untuk pensiun dan menyerahkan pengelolaan usaha kepada anak tertuanya hingga saat ini. Optik yang masih bertahan hingga saat ini adalah optik yang beralamat di Jalan Sukajadi. Optik yang melayani peserta ASKES maupun BPJS yang berlaku saat ini.

2.1.2 Logo Perusahaan

Berikut ini adalah logo dari Optik Yuda yang dapat dilihat pada gambar 2.1.



Gambar 2. 1 Logo Optik Yuda

2.1.3 Unit Kerja

Di dalam lingkungan perusahaan Optik Yuda terdapat empat unit kerja utama yaitu owner, sales, facet, dan admin. Deskripsi dari masing-masing unit kerja yang ada di Optik Yuda adalah sebagai berikut:

1. Owner : Sebagai pemilik sekaligus direktur, memegang dan mengendalikan seluruh kebijakan di dalam perusahaan.
2. Sales : Menawarkan dan menjual produk yang ada di optik, melayani segala keperluan ataupun keluhan dari konsumen.
3. Facet : Bertanggung jawab atas pemotongan dan pemasangan lensa. Melakukan pembelanjaan lensa, memeriksa stok lensa yang masuk dan keluar.
4. Admin : Bertanggung jawab terhadap segala pembukuan pemasukan dan pengeluaran, membuat laporan bulanan pembelanjaan lensa dan frame, membuat laporan omzet dan rugi-laba, melakukan pemesanan lensa ke pabrik, dan juga bertugas mengajukan klaim BPJS setiap bulannya.

2.1.4 Data Teknis Perusahaan

Nama Perusahaan	:	Optik Yuda
Alamat Perusahaan	:	Komplek ruko sukajadi no. 18 D
No Telp	:	022-2040008
Jenis Usaha	:	OPTIK
Tahun Didirikan	:	1981
No SIUP	:	No. 510/1-0552/2007/3575-bppt/2010
Izin Dinkes	:	No. 445/ 9826-dinkes/07-SI-Opt/XII/11

2.2 Landasan Teori

Sub bab ini berisi tentang teori-teori pendukung yang digunakan dalam proses analisis dan implementasi dalam tugas akhir ini.

2.2.1 Pengertian Data

Data dan Informasi mempunyai keterkaitan antara satu dengan yang lainnya. Data adalah fakta-fakta atau pesan-pesan yang belum dievaluasi atau bahan mentah dari informasi yang melalui pengolahan tertentu dibentuk menjadi suatu informasi. Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian nyata.

Data merupakan suatu bentuk keterangan-keterangan yang belum diolah atau dimanipulasi sehingga belum begitu berarti bagi sebagian pemakai. Sedangkan informasi merupakan data yang telah diolah sehingga memiliki arti bagi yang membutuhkan.

2.2.2 Konsep Basis Data

Basis data dapat didefiniskan 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 bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan [9].

“ 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/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan. "

Jadi secara umum basis data dapat diartikan sebagai kumpulan data yang tersimpan dalam suatu tabel yang saling berelasi. Adapun kegunaan dari basis data adalah:

1. Basis data merupakan komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi yang akurat, tepat pada waktunya, dan relevan dimana informasi dikatakan bernilai jika manfaanya lebih efektif dibandingkan dengan biaya untuk mendapatkannya.
3. Mengurangi duplikasi data (*data redudancy*). Meningkatkan relasi antar data (*data relatability*).
4. Mengurangi pemborosan *external storage*.

Di dalam basis data terdapat beberapa istilah berikut yang juga dikenal sebagai urutan jenjang data.

1. *Characters*

   Merupakan bagian data yang terkecil berupa karakter numerik, huruf, maupun karakter spesial (simbol) yang membentuk item data / *field*
2. *Field*

   Merepresentasikan suatu atribut dari *record* yang menunjukan suatu item dari data, misalnya nama.
3. *Record*

   Merupakan kumpulan dari *field*, menggambarkan suatu unit data individu tertentu. Kumpulan *record* membentuk suatu *file*. Misalnya, *file* personalia, tiap-tiap *record* mewakili data tiap-tiap karyawan.
4. *File*

   File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya, *file* mata kuliah berisi data tentang semua mata kuliah yang ada.

5. *Database*

   Merupakan kumpulan dari *file* / tabel.

2.2.2.1 Sistem Basis Data

Sistem basis data adalah merupakan suatu sistem yang terdiri atas kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi *file-file* tersebut [9]. Komponen utama dari sistem basis data adalah:

1. Data yang disimpan dalam basis data.
2. *Hardware*: *storage*, *processor*, *memory*.
3. *Software DBMS*, *Report-writer*, *design arts*, dll.

Terdapat beberapa kategori pengguna sistem basis data berdasarkan kemampuannya yaitu Pengguna Awam (*Naïve User*), Pengguna Biasa (*Casual User*), *Programmer*, dan *Administrator*.

2.2.2.2 Bahasa dalam Sistem Basis Data

Ada beberapa bahasa yang digunakan dalam sistem basis data diantaranya adalah sebagai berikut:

1. *Data Definition Language*

Data Definition Language (DDL) ini berfungsi untuk menspesifikasikan skema basis data. Dengan bahasa ini hal-hal yang dapat dilakukan adalah membuat tabel baru, membuat indeks, mengubah struktur tabel, menentukan struktur penyimpanan tabel, dan lain-lain. Hal yang dasar yang dapat dilakukan DDL adalah menciptakan, mengubah, dan menghapus basis data.

2. *Data Manipulation Language*

Data Manipulation Language (DML) adalah perintah-perintah yang digunakan untuk mengubah, memanipulasi dan mengambil data pada basis data. Tindakan seperti menghapus, mengubah, dan mengambil data menjadi bagian dari DML. DML dibagi atas 2 jenis:

a. Prosedural

Prosedural menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkannya.

b. Non-Prosedural

Nonprosedural menuntut pengguna menentukan data apa yang diperlukan tetapi tidak perlu menyebutkan cara mendapatkannya.

3. *Transaction Control*

Transaction Control adalah bahasa basis data yang mengatur transaksi yang dilakukan oleh *Data Manipulation Language* (DML). *Transaction Control* ini memiliki peran yang sangat besar untuk menentukan dilakukan atau tidaknya perubahan-perubahan data yang ada pada basis data. Contoh dari transaction control adalah perintah *commit* dan *rollback.*

2.2.2.3 Sistem Manajemen Basis Data

Sistem Manajemen Basis Data atau *Database Management System* (DBMS) adalah perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar [4].

Adapun keuntungan dari DBMS (*Database Manegement System*) adalah:

1. *Data independence*, DBMS dapat mengurangi kebergantungan antar data.
2. *Efficient data access*, DBMS dapat menimpa data dan me-*retrive* data dengan efisien. Apabila suatu saat data tersebut dibutuhkan, maka efektif dalam pemanggilannya.
3. *Data integrity and security*, saat data diakses menggunkan DBMS, DBMS mengintergrasi *constraints* dari tiap data tersebut.
4. *Data administration*, Administrasi data digunakan untuk mengatur hal-hal yang berhubungan dengan distribusi data, hak akses user yang mendistribusikan atau menggunakan data tersebut, mengorganisasikan

data untuk mengurangi redudansi data dan *fine-tuning* penyimpanan data sehingga *retrive* data berjalan dengan efisien.

5. *Concurrent access and crash recovery*, DBMS dapat menjadwalkan *concurrent access* (pengaksesan secara bersamaan) terhadap suatu data sehingga menghindari dari *crash*.
6. *Reduced application development time*, DBMS mendukung banyak fungsi penting yang diakses oleh banyak aplikasi yang tersimpan dalam media penyimpanan.

2.2.2.4 Sistem Manajemen Basis Data Relasional

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

Ada tiga prinsip dalam RDBMS:

1. *Data Definition*
   Mendefinisikan data yang akan dibuat, seperti nama tabel dan *field* serta keterangan mengenai *field* tersebut.
2. *Data Manipulation*
   Memanipulasi data yang sudah dibuat, seperti mengubah nama *field* dan keterangannya, menghapus *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 yang memiliki 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 pemakai, termasuk operasi-operasi yang dapat dilakukan.
6. Spesifikasi batasan integrasi.

2.2.3 Data Warehouse

Data warehouse merupakan kumpulan informasi logikal, yang dikumpulkan dari berbagai operasional *database* yang mendukung aktivitas para bisnis analis dalam pekerjaan mengambil keputusan. *Data warehouse* menggambarkan hal yang mendasar perbedaan cara pandang tentang mengorganisasi dan mengatur informasi dalam suatu organisasi.

Data warehouse adalah multidimensional dalam relasional model *database*, informasi ditentukan dalam suatu rangkaian dari tabel yang berdimensi dua. Banyak *data warehouse* adalah *multi dimensional*, artinya bahwa *data warehouse* terdiri dari *layer-layer*, kolom dan baris. *Layer-layer* dalam suatu *data warehouse* menunjukkan informasi yang disampaikan pada dimensi yang berbeda. *Data warehouse* merupakan kombinasi dari *database* yang berbeda-beda, *data warehouse* mengkombinasikan informasi dengan meringkas *(summarizing*) dan mengelompokkan (*aggregation*). Pada saat mengekstrak informasi dari berbagai operasional *database* untuk membuat suatu *data warehouse*, pengumpulan informasi yang diperlukan untuk pembuat keputusan. Informasi yang diperlukan ini didefinisikan oleh persetujuan para pengguna sesuai dengan informasi yang mereka butuhkan dalam pengambilan keputusan. Jadi suatu *data warehouse* hanya berisi informasi yang relevan dengan kebutuhan user untuk mendukung pengambilan keputusan.

“ *Menurut Paul Lane, data warehouse merupakan database relasional yang dirancang lebih untuk query dan analisis proses transaksi, biasanya mengandung sejarah data transaksi dan mungkin juga data dari sumber lain. Data warehouse*

memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi untuk menggabungkan / konsolidasi data dari berbagai sumber. ‘‘

Secara umum *data warehouse* merupakan metode dalam perancangan basis data, yang menunjang DSS (*Decission Support System*) dan EIS (*Executive Information System*). Dari definisi-definisi yang telah dijelaskan, dapat disimpulkan *data warehouse* adalah basis data yang saling bereaksi yang dapat digunakan untuk *query* dan analisisis, bersifat orientasi subjek, terintegrasi, *time variant*, tidak berubah yang digunakan untuk membantu para pendukung keputusan.

2.2.4 Data *Mart*

Data mart merupakan bentuk sederhana dari *data warehouse* yang difokuskan pada subjek tunggal (atau area fungsional) contohnya seperti Bagian Penjualan, Bagian Keuangan, atau Bagian Pemasaran. Biasanya data *mart* dibangun dan dikendalikan oleh salah satu departemen dalam sebuah orgaanisasi. Dikarenakan *data mart* fokus terhadap subjek tunggal maka *data mart* akan mengambil data hanya dari beberapa sumber seperti sistem operasional internal, *data warehouse*, atau data eksternal.

Ada empat tugas yang bisa dilakukan dengan adanya data *mart*, keempat tugas tersebut adalah sebagai berikut [1]:

1. Pembuatan Laporan

Pembuatan laporan merupakan salah satu kegunaan *data mart* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan per hari, per bulan, per tahun atau jangka waktu kapanpun yang diinginkan untuk menunjang proses pendukung keputusan.

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

Dengan adanya *data mart*, semua informasi baik detail maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah didapat. OLAP menggunakan konsep *multi dimensional* data dan memungkinkan pengguna menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL. Hal

ini dimungkinkan karena pada konsep *multi dimensional*, maka data yang berupa fakta yang sama bisa dilihat dengan menggunakan fungsi yang berbeda.

3. *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 pelakunya.

4. 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 diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan. Informasi dan data pada laporan *data mart* menjadi target informatif bagi pengguna.

2.2.4.1 Perbedaan Data *Mart* dan Data *Warehouse*

Tidak seperti *data mart*, *data warehouse* menangani beberapa subjek area dan biasanya diterapkan dan dikendalikan oleh organisasi terpusat seperti perusahaan yang berbasis *Information Technology* (IT). Seringkali hal itu disebut *data warehouse* yang terpusat atau *enterprise data warehouse*. Biasanya *data warehouse* mengumpulkan data dari beberapa sistem.

Tidak ada ketentuan yang membatasi ukuran dari *data mart* ataupun komplekstitas dari *decision-support* data yang terdapat di dalam *data mart*. Meskipun begitu, *data mart* biasanya lebih kecil dan komplekstitasnya kurang dari *data warehouse*. Oleh karena itu *data mart* lebih mudah untuk dibangun dan dipelihara. Perbedaan mendasar antara *data warehouse* dan *data mart* dapat dilihat pada tabel 2.1 berikut:

Tabel 2. 1 Perbedaan Antara Data Warehouse dan Data Mart

Kategori	**Data Warehouse**	**Data Mart**
Cakupan	Perusahaan	Bagian Tertentu / Divisi Tertentu
Subjek	Beberapa Subjek	Subjek Tunggal
Sumber Data	Banyak	Lebih Sedikit
Ukuran	100 GB-TB+	<100 GB
Waktu Pelaksanaan	Bulan Hingga Tahun	Bulan

2.2.4.2 Dependent dan Independent Data Mart

Terdapat dua jenis *data mart*: *dependent* dan *independent*. Kategori tersebut berdasarkan pada sumber data yang diambil oleh *data mart*. *Dependent data mart* mengambil data dari sebuah *data warehouse* yang telah dibuat. Sedangkan *independent data mart* adalah sistem *standalone* yang dibangun dengan mengambil data langsung dari data operasional atau data eksternal atau keduanya [3].

Perbedaan utama antara *dependent* dan *independent data mart* adalah bagaimana *data mart* akan diisi dan bagaimana mendapatkan data dari sumber lalu mengisinya ke dalam *data mart*. Langkah ini disebut proses ETL (*Extract Transform Load*). Proses ini melibatkan pemidahan data dari sistem operasional, menyaringnya, dan memasukannya ke *data mart* [1].

Dalam *dependent data mart* prosesnya akan lebih sederhana karena data yang telah diformat dan diringkas telah dimuat ke dalam *data warehouse*. Proses ETL dalam dependent *data mart* sebagian besar adalah proses identifikasi subset data apa yang tepat dan relevan sesuai dengan subjeknya dan setelah itu *copy* dari subset data akan berpindah ke *data mart* dalam bentuk yang sudah diringkas.

Dalam *data mart independent* akan melakukan semua aspek dari proses ETL, seperti halnya *data warehouse*. Namun jumlah sumber data akan lebih sedikit dan jumlah data yang terkait dengan *data mart* akan kurang dari *data warehouse*, mengingat fokus *data mart* adalalah pada subjek tunggal.

Alasan di balik pembangunan kedua jenis *data mart* biasanya akan berbeda. *Dependent data mart* biasanya dibangun untuk peningkatan kinerja dan ketersediaan, kontrol yang lebih baik, dan biaya komunikasi yang lebih rendah karena setiap departemen mempunyai *data mart* masing-masing. Pembangunan *independent data mart* sering didorong oleh kebutuhan untuk mendapatkan solusi dalam waktu yang lebih singkat.

2.2.5 Karakteristik Data Mart atau Data Warehouse

Karakteristik *data warehouse* menurut Inmon [5] yaitu:

1. *Subject Oriented*

Data warehouse berorientasi subjek artinya *data warehouse* di desain untuk menganalisa data berdasarkan subjek-subjek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu. *Data warehouse* diorganisasikan disekitar subjek-subjek utama dari perusahaan (*customers*, *products* dan *sales*) dan tidak diorganisasikan pada area aplikasi utama (*customer invoicing, stock control dan product sales*). Hal ini dikarenakan kebutuhan dari *data warehouse* untuk menyimpan data-data yang bersifat sebagai penunjang suatu keputusan, dari pada aplikasi yang berorientasi terhadap data. Jadi dengan kata lain, data yang disimpan adalah berorientasi kepada subjek bukan terhadap proses.

2. *Integrated*

Data Warehouse dapat menyimpan data yang berasal dari sumber-sumber yang terpisah kedalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Dengan demikian data tidak bisa dipecah-pecah karena data yang ada merupakan suatu kesatuan yang menunjang keseluruhan konsep *data warehouse* itu sendiri. Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara sepeti

konsisten dalam penamaan *variable*, konsisten dalam ukuran *variable*, konsisten dalam struktur pengkodean dan konsisten dalam atribut fisik dari data.

3. *Time Variant*

Seluruh data pada data *warehouse* dapat dikatakan akurat atau valid pada rentang waktu tertentu. Untuk melihat interval waktu yang digunakan dalam mengukur keakuratan suatu *data warehouse*, kita dapat menggunakan cara antara lain:

a. Cara yang paling sederhana adalah menyajikan *data warehouse* pada rentang waktu tertentu, misalnya antara 5 sampai 10 tahun ke depan.
b. Cara yang kedua, dengan menggunakan variasi atau perbedaan waktu yang disajikan dalam *data warehouse* baik *implicit* maupun *explicit* secara *explicit* dengan unsur waktu dalam hari, minggu, dan bulan. Secara *implicit* misalnya pada saat data tersebut diduplikasi pada setiap akhir bulan, atau per tiga bulan. Unsur waktu akan tetap ada secara *implicit* didalam data tersebut.
c. Cara yang ketiga, variasi waktu yang disajikan *data warehouse* melalui serangkaian *snapshot* yang panjang. *Snapshot* merupakan tampilan dari sebagian data tertentu sesuai keinginan pemakai dari keseluruhan data yang ada bersifat *read-only*.

4. *Non Volatile*

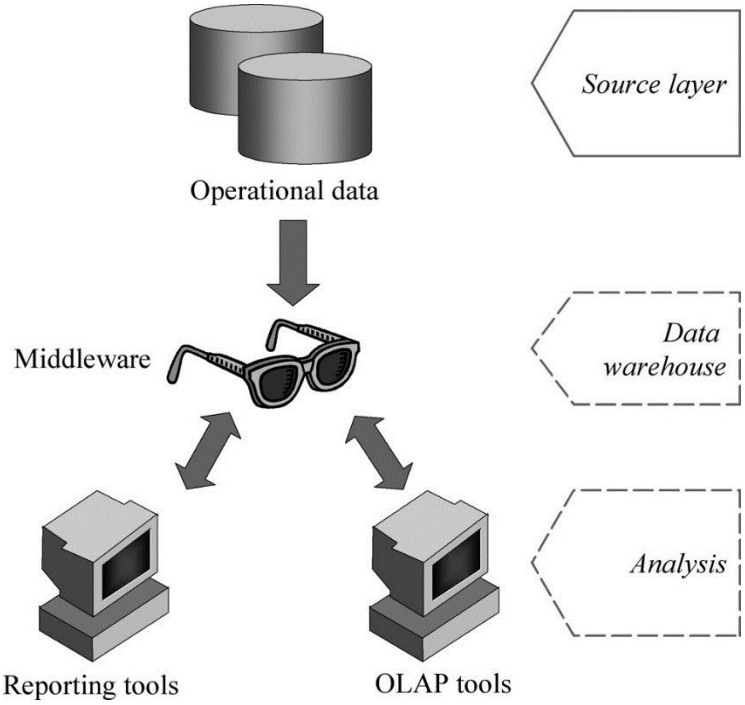
Karakteristik keempat dari *data warehouse* adalah *non-volatile*, maksudnya data pada *data warehouse* tidak di-*update* secara *real time* tetapi di *refresh* dari sistem operasional secara reguler. Data yang baru selalu ditambahkan sebagai suplemen bagi *database* itu sendiri dari pada sebagai sebuah perubahan. *Database* tersebut secara *continue* menyerap data baru ini, kemudian secara *incremental* disatukan dengan data sebelumnya.

2.2.6 Arsitektur Data Warehouse

Arsitektur *Data Warehouse* atau *Data mart* menurut Matteo Golfarelli dan Stefano Rizzi dalam bukunya yang berjudul *Data Warehouse Design: Modern Principles and Methodologies*, mengelompokkan arsitektur *data warehouse* menjadi 3 kelompok, terdiri dari *single layer architechture, two layer architecture*, dan *three layer architecture*.

1. *Single Layer Architercture*

Single Layer Architechture pada umumnya tidak sering digunakan dalam suatu kasus. Tujuannya adalah untuk meminimalkan jumlah data yang disimpan, untuk mencapai tujuan ini, ia bisa menghilangkan redudansi data. Gambar 2.1 menunjukkan hanya lapisan fisik yang tersedia. Ini berarti bahwa *data warehouse* di implementasikan sebagai pandangan multidimensi data operasional yang dibuat oleh *middleware* tertentu, ataulapisan pengolahan menengah. *Middleware* merupakan komponen perantara yang memungkinkan *client* dan (lapisan aplikasi dan sistem operasi) saling terhubung dan berkomunikasi satu sama lain.



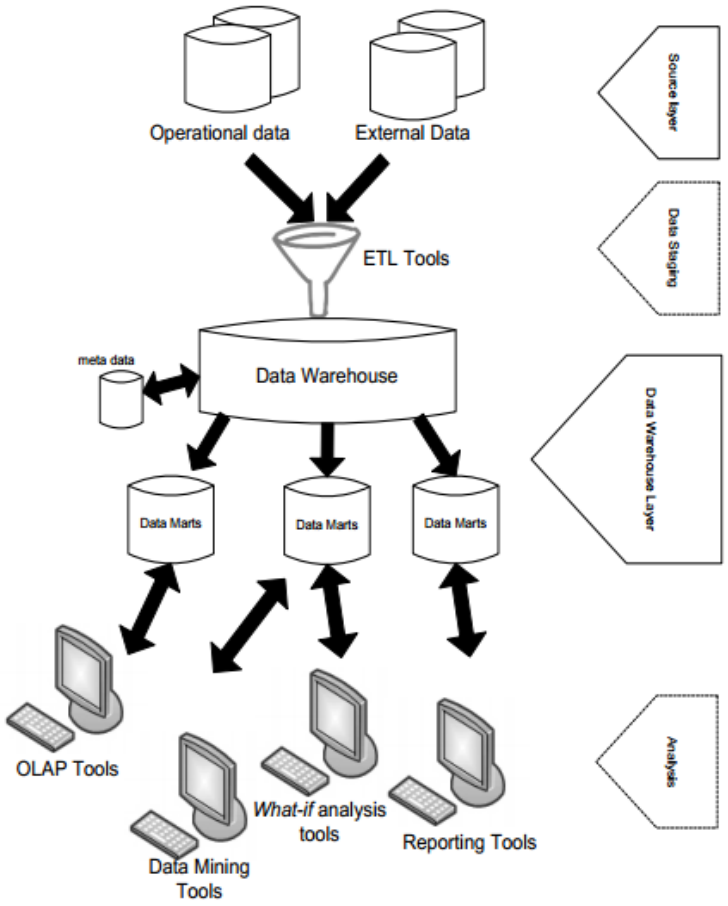
Gambar 2. 2 *Single Layer Architecture* [1]

Kelemahan arsitektur ini terletak pada kegagalan untuk memenuhi persyaratan untuk pemisahan antara pengolahan analisis dan transaksional. Gambar 2.1 *Single Layer Architechture Query* analisis yang disampaikan kepada data operasional setelah *middleware* menganalisis. Untuk alasan ini, pendekatan *virtual* untuk *data warehouse* dapat berhasil hanya jika analisis kebutuhan utama dibatasi dan volume data untuk tujuan analisis sangat besar.

2. *Two Layer Architecture*

Two layer architechture yang di tunjukan pada Gambar 2.2, merupakan arsitektur dari *data warehouse* yang menggunaka dua *layer* atau lapisan sebagai proses pemisahan antara sumber yang tersedia dengan *data warehouse*. Arsitektur ini memiliki empat tahap aliran data, yaitu:

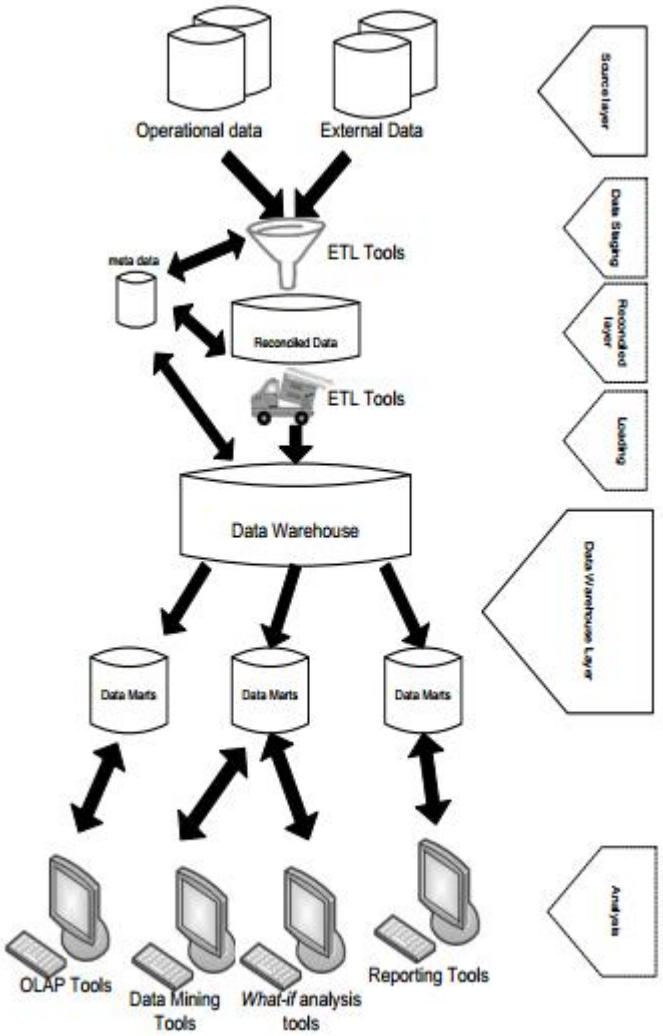
a. *Source Layer*, Sebuah sistem *data warehouse* menggunakan sumberdata heterogen. Data yang awalnya disimpan ke *database* relasional perusahaan atau mungkin berasal dari sistem informasi perusahaan.
b. *Staging Data*, Data yang disimpan ke sumber harus diekstrak, kemudian dibersihkan untuk menghilangkan inkonsistensi, dan terintegrasi untuk menggabungkan sumber yang heterogen menjadi satu skema umum. Skema umum tersebut adalah Ekstraksi, Transformasi, dan Loading (ETL).
c. *Data Warehouse Layer*, pada lapisan ini data dapat diakses secara langsung, tetapi juga dapat digunakan sebagai sumber untuk membuat *data mart* dan dirancang untuk departemen perusahaan tertentu. *Meta data* sebagai *repository* yang menyimpan informasi pada sumber, *staging data*, dan sebagainya.
d. *Analysis Layer*, data yang terintegrasi secara efisien dan fleksibel diakses untuk mencetak laporan, menganalisis informasi yang dibutuhkan dan mensimulasikan hipotesis skenario bisnis yang membahas berbagai jenis pengambilan keputusan perusahaan.



Gambar 2. 3 *Two Layer Architecture* [1]

3. *Three Layer Architecture*

Dalam arsitektur ini, lapisan ketiga adalah lapisan penyesuaian data (*reconciled data*). Lapisan ini terwujud dari data operasional yang diperoleh setelah mengintegrasikan dan membersihkan sumber data. Akibatnya data tersebut adalah terintegrasi, konsisten, dan rinci. Gambar 2.3 menunjukkan sebuah *data warehouse* yang tidak di dapat dari sumbernya secara langsung, tetapi diperoleh dari *reconciled data*.



Gambar 2. 4 *Three Layer Architecture* [1]

2.2.7 Multi Dimensinal Modeling

Multi Dimensional Modeling adalah teknik untuk memvisualisasi model data sebagai suatu kumpulan dari ukuran yang dideskripsikan dengan aspek-aspek bisnis [3]. 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 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. Dalam *data mart*, *fact* di implementasikan dalam tabel dasar dimana semua data numerik dan disimpan.

2. *Dimensions*

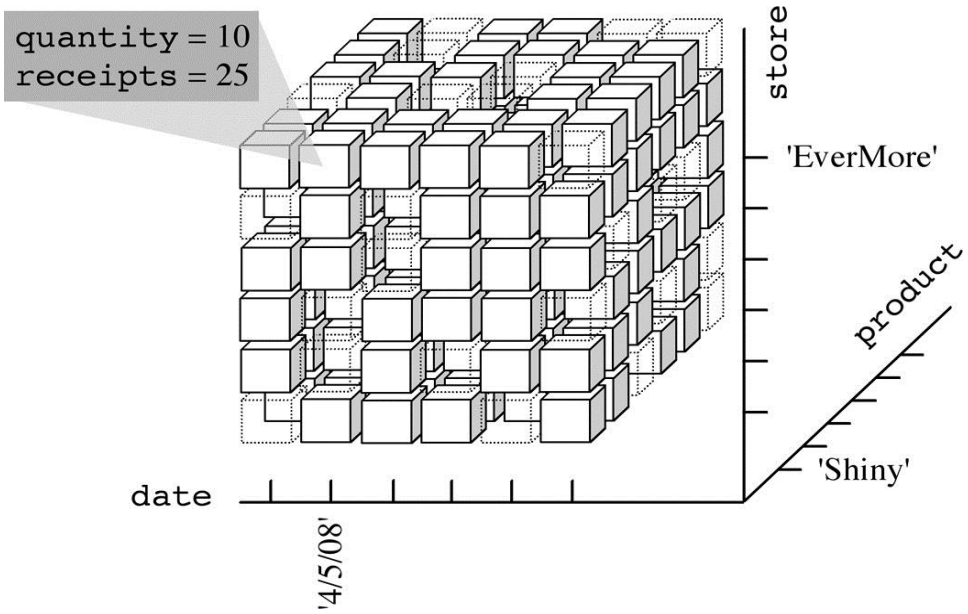
Dimensions adalah suatu koleksi dari anggota atau unit-unit data dengan tipe yang sama. Dalam sebuah diagram, suatu dimensi biasanya direpresentasikan dengan suatu axis. Dalam *dimensional* model, semua data menunjukan *fact* table yang diasosiasikan dengan satu dan hanya satu member setiap *multiple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

3. *Measure*

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

2.2.8 Visualisasi Dimensional Model

Cara yang paling populer dalam memvisualisasikan suatu model *dimensional* adalah dengan menggambarkan sebuah *cube* data. Contohnya dapat dilihat pada gambar 2.4 yang dapat menggambarkan tiga model *dimensional* menggunakan sebuah kubus. Biasanya suatu dimensional model terdiri dari lebih tiga dimensi dan digambarkan sebagai suatu *hypercube*, akan tetapi *hypercube* sulit untuk divisualisasikan, jadi sebuah kubus lebih biasa digunakan. Gambar berikut merupakan *measurement* adalah volume dari pemesanan konsumen, yang mana dijelaskan dengan kombinasi dari tiga dimensi yaitu lokasi, produk dan waktu.

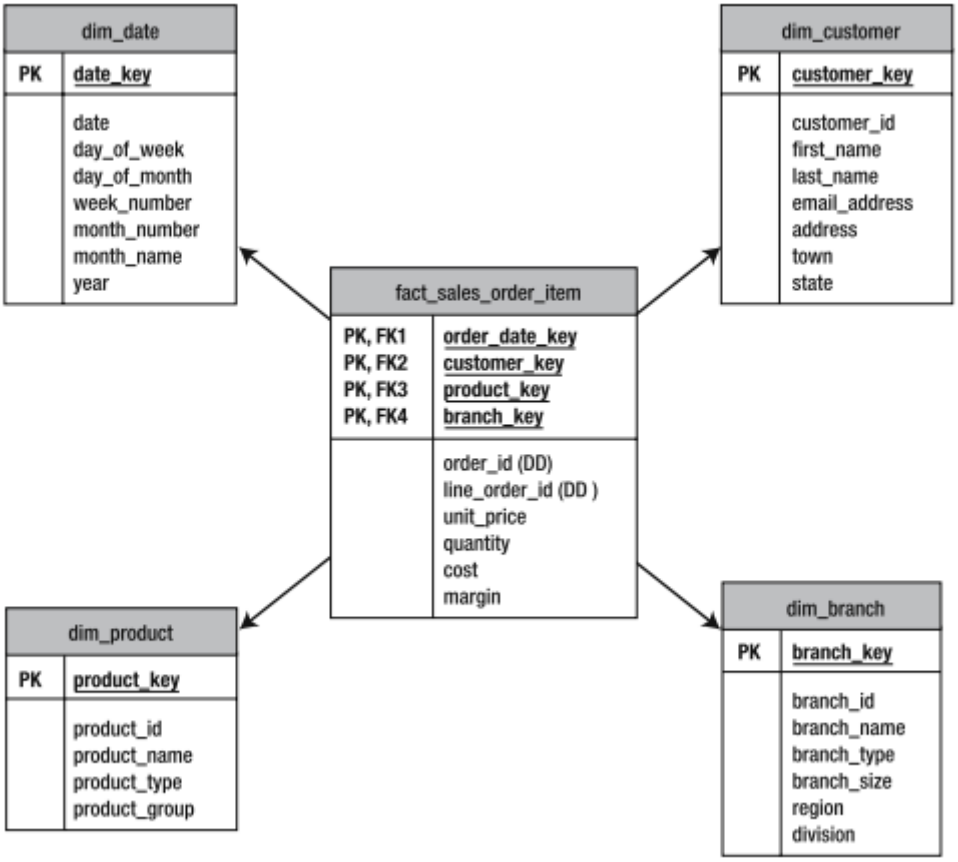


Gambar 2. 5 Visualisasi Multi Dimensional Model Dengan Kubus [1]

Terdapat tiga konsep yang dapat memodelkan lebih jelas dalam penentuan *fact table*, measure, dan dimensi, yaitu:

1. *Star Schema*

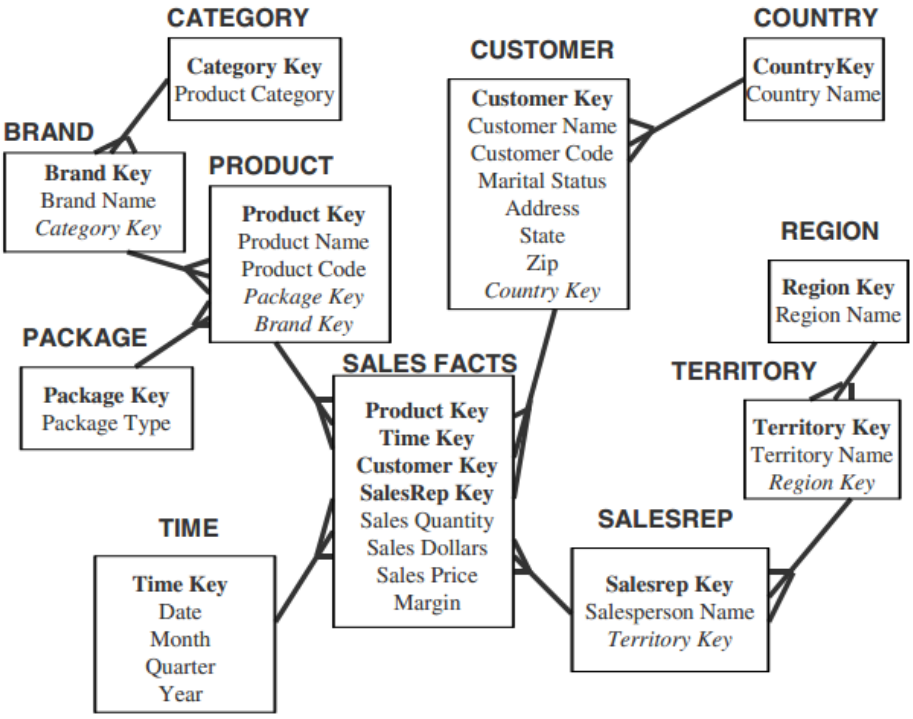
Star Schema adalah model dasar dari pemodelan multidimensi yang dapat di lihat pada gambar 2.5. Model *star* memiliki satu tabel induk yang dinamakan *fact table* dan kumpulan dari tabel-tabel kecil yang disebut tabel dimensi, disusun dalam pola-pola melingkar mengelilingi table *fact*.



Gambar 2. 6 Star Schema [6]

2. *Snowflake Schema*

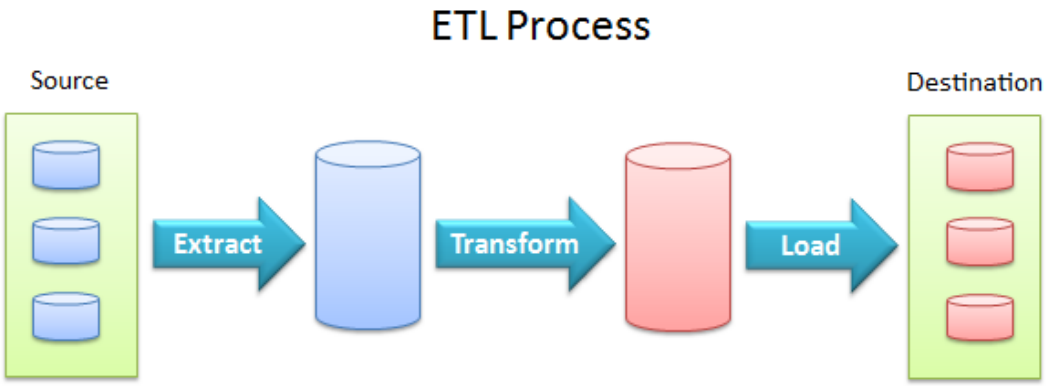
Selain *star schema* dalam menampilkan desain sebuah data *mart* juga dapat menggunakan model *Snowflake* sebagai contoh dapat dilihat pada 2.6. Dalam sebuah model *snowflake* terdapat sebuah *fact table* yang dikelilingi oleh beberapa *dimension table*. Namun *dimension table* itu sendiri dapat seolah-olah berupa sebuah *fact table* lain yang juga memiliki dimensinya sendiri.



Gambar 2. 7 Snowflake Schema [7]

2.2.9 Extract Transform Load (ETL)

Proses ETL (*Extract-Transform-Load*) terjadi mulai dari ketika data memasuki hingga keluar dari *Staging Area*. Pada tahap ini, data diekstrak (*Extrac*t) dari sistem OLTP dan dikirimkan ke *Staging Area*. *Staging Area* adalah lokasi penempatan sementara untuk data sumber yang sudah dirubah formatnya (*Transform*), untuk dipersiapkan ke proses selanjutnya. Pada *Staging Area*, bentuk data dapat berupa sekumpulan *file* teks baku (*flat file*s), tabel *database*, atau keduanya. Data terpadu dengan data lain dan dirubah ke dalam bentuk yang lebih homogen. Selanjutnya data dimuat (*Load*) ke tabel-tabel data *mart* yang ditentukan. Secara umum proses ETL dapat dilihat pada gambar 2.7



Gambar 2. 8 Proses ETL

Berikut ini adalah penjelasan dari tiap-tiap proses *extract*, *transform*, dan *load*.

1. *Extract*

Bagian pertama dari suatu proses ETL aalah mengekstrak dari sumber data disebut ekstrak, karena proses pengambilan data ini tidak mengambil keseluruhan data yang ada di *database* operasional, melainkan hanya mengambil data matang saja. Proses ini meliputi penyaringan data yang akan digunakan dalam pembuatan *data warehouse*. Dapat langsung dimasukan dalam *data warehouse* atau dimasukan dalam tempat penampungan sementara terlebih dahulu

2. *Transform*

Proses yang ke dua adalah transformasi data yang telah diekstrak ke dalam format yang diperlukan. Hal ini perlu dilakukan mengingat data yang diambil bersal dari sumber yang berbeda yang kemungkinan memiliki standarisasi yang berbeda pula. Data dari beberapa sistem perlu ditransforamsikan ke dalam format umum yang disepakati dan digunakan dalam *data warehouse*, seperti :bisa jadi untuk suatu item yang sama memiliki dua nama yang berbeda, misalnya nama produk dengan kode :1234: disebut “roti” sedangkan kode yang sama “1234” ditempat lain disebut “kue”.

3. *Load*

Tahap *load* adalah menload data ke dalam target akhir yang pada umumnya adalah *data warehouse*. Bergantung apda kebutuhan organisasi, proses ini bervariasi secara luas. Beberapa *data warehouse* memperbolehkan melakukan penulisan informasi yang ada secara kumulatif, dengan data yang diperbaharui tiap minggu.

2.2.10 On-line Analytical Processing (OLAP)

Online analytical processing (OLAP) merupakan salah satu *tools* yang digunakan untuk mengakses informasi dalam *data warehouse* atau *data mart*. Teknologi OLAP memungkinkan *data warehouse* digunakan secara aktif untuk proses onlineanalisis, memberikan respon yang cepat terhadap analytical queries yang kompleks [8].

Data model *multi dimensional* dan teknik agregasi data yang dimiliki oleh OLAP dapat mengatur dan membuat kesimpulan dari data dalam jumlah besar sehingga dapat di evaluasi secaa cepat dengan menggunakan *online analysis* dan *graphical tool*. Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk men-*support* proses analisis secara *real time*. Berikut ini adalah karakteristik dari OLAP:

a. Menggunakan teknik analisis *multi dimensional* data.

b. Menyediakan dukungan *database* tingkat lanjut.

c. Menyediakan cara pakai yang mudah dan *user interface* yang mudah dipahami.

d. Mendukung arsitektur *client-server*.

2.2.10.1 OLAP Service

OLAP *services* adalah suatu yang menggambarkan beberapa perangkat yang bekerja bersama-sama untuk membantu dalam menganalisa data yang ada. Informasi dari *data warehouse* atau *data mart* di ekstrak secara periodik dan digunakan untuk mengupdate objek yang ada dalam OLAP *service*. OLAP *server*

mengambil data dari *data warehouse* atau *data mart* dan *relational* OLAP serta meng*update* informasi yang disimpan dalam *multidimensiona*l OLAP (MOLAP). Setelah informasi dikumpulkan maka *pivot table service* yang bekerja sama dengan *excel pivot table* atau perngkat lain yang mendukung OLE DB dengan ekstensi OLAP dapat melakukan akses dan memanipulasi data yang ada [8].

2.2.10.2 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 [8]:

1. Multidimensional OLAP (MOLAP)

MOLAP merupakan cara tradisional dalam analisis OLAP. Dalam MOLAP data disimpan dalam bentuk *Multidimensional Database*. Tidak seperti basis data relasional yang menyimpan data dalam jumlah baris dalam tabel, sebuah data *multidimensional* menyimpan data 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 MOLAP juga mengandung agregasi data sehingga dapat memberikan respon yang cepat terhadap *query*.

2. Relational OLAP (ROLAP)

ROLAP merupakan cara lain yang digunakan untuk melakukan partisi menggunakan tabel relasional dalam *data warehouse*. 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 data OLAP. Hal ini hanya berlaku pada tingkat tertentu saja, pada sebuah basis data yang terdiri dari ribuan atau ratusan ribu *record*s maka menampilkan data OLAP akan menjadi sebuah masalah, karena banyak data yang harus di*query*. Dan hal ini lah yang menjadi keterbatasan partisi ROLAP.

3. Hybrid OLAP (OLAP)

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*.

2.2.10.3 Teknik Dalam OLAP

OLAP meiliki berbagai macam teknik yaitu teknik *drill-down and roll-up*, *slice and dice*, dan *pivoting*. Berikut ini adalah penjelasan dari masing-masing teknik:

1. *Drill-down and Roll-up*

Drill-down and roll-up adalah operasi untuk melihat data global atau detail di sepanjang level hirarki dimensi. *Roll-up* untuk melihat data secara global atau rangkuman (*summary*). *Drill-down* memandu pengguna untuk memperoleh data yang lebih detail. *Drill-down* ini biasa digunakan untuk menjawab pertanyaan atas suatu kasus tertentu. Misalnya untuk menjawab pertanyaan ketika sebuah *summary number* (rata-rata atau jumlah) di bawah atau di atas harapan.

2. *Slice and Dice*

Slice and dice adalah operasi untuk melihat data sebagai visualisasi dari kubus. Dengan *slice dan dice* pengguna dapat melihat data dari beberapa perspektif. Pengguna dapat mengekstrak bagian dari data *agregrated* dan dapat memeriksa dengan detail berdasarkan dimensi-dimensi yang diinginkan. *Data Agregrated* merupakan data pra-perhitungan (*precalculated*) dalam bentuk rangkuman data (*data summarized*) sehingga *query* pada kubus (*cube*) lebih cepat. *Slice* memotong kubus sehingga dapat memfokuskan pada perspektif yang spesifik (pada suatu dimensi). Sedangkan *dice* memberikan kemampuan untuk melihat pemilihan data pada dua dimensi atau lebih, yaitu dengan merotasi *cube* pada perspektif yang lain sehingga pengguna dapat melihat lebih spesifik terhadap data yang di analisis.

3. *Pivotting*

Pivotting merupakan suatu teknik untuk saling menukarkan dimensi data. Dengan melakukan *pivotting* maka dapat diamati suatu informasi atau data dari sudat pandang yang berbeda, sehingga diharapkan akan dapat memperjelas analisis yang dilakukan.

2.2.11 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.

2.2.12 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.

2.2.13 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 strore dalam suatu *entity relationship diagram* (ERD).

2.2.14 Visualisasi Data

Visualisasi adalah rekayasa dalam pembuatan gambar, diagram, atau animasi untuk menampilkan suatu informasi tertentu. Visualisasi adalah konversi data ke dalam format visual atau tabel sehingga karakteristik dari data dan relasi diantara item data atau atribut dapat dianalisi atau dilaporkan. Tujuan utama dari visualisasi data adalah untuk mengkomunikasikan informasi secara jelas dan efisien kepada pengguna melalui grafik informasi yang dipilih, seperti tabel dan grafik. Visualisasi yang efektif dapat membantu pengguna dalam menganalisis data. Umumnya visualisasi yang sering digunakan dalam bentuk tabel dan grafik.

1. Tabel

Tabel merupakan kumpulan data yang disusun berdasarkan baris dan kolom. Baris dan kolom berfungsi untuk menunjukan keterkaitan antar data. Tabel merupakan alat bantu visual yang paling umum digunakan.

2. Bar Chart

Bar chart atau biasa dikenal sebagai *bar graph* (dalam bahasa Indonesia disebut diagram batang) merupakan salah satu jenis grafik yang paling umum [11]. *Bar chart* biasanya digunakan untuk membandingkan sesuatu antara kelompok data yang beberda atau mengetahui perubahan dari waktu ke waktu. Keuntungan dalam menggunakan *bar chart* adalah kemudahan dalam membandingkan tiap-tiap variabel, hubungan antara tampilan vertikal dan horizontal mudah dipahami, dan efektif dalam menyajikan suatu trend atau perubahan dari waktu ke waktu.

Bar chart adalah grafik yang efektif untuk membandingkan sesuatu diantara kelompok yang beberda. Bar chart menampilkan perbandingan angka berdasarkan waktu yang diinginkan misalnya perbandingan data penjualan dalam setiap bulan atau setiap tahun.

Bar chart biasanya terdiri dari dua sumbu utama yaitu sumbu x (horizontal) dan sumbu y (vertikal). Dimana sumbu x merepresentasikan kategori atau waktu. Sedangkan untuk sumbu y merepresentasikan kuantitas.

3. Pie Chart

Pie chart adalah sebuah grafik bundar yang menunjukan data dan informasi yang terbagi-bagi kedalam potongan-potongan. Setiap potongan merupakan bagian dari keseluruhan data, jumlah nilai dari *pie chart* selalu 100 persen. *Pie chart* biasanya digunakan untuk melakukan perbandingan, dimana dalam *pie chart* akan terlihat jelas mana yang lebih mendominasi dan tidak. Efektifnya *pie chart* menampilkan 6 kategori data atau kurang, jika semakin banyak nilai yang dibandingkan di dalam *pie chart* maka semakin kecil potongan-potongan yang terbentuk dan akan semakin sulit untuk dilihat.

4. Line Chart

Line chart atau biasa dikenal sebagai *line graph* adalah tipe grafik yang mevisualisasikan suatu nilai di waktu tertentu. *Line chart* biasanya terdiri dari dua sumbu utama yaitu sumbu x (horizontal) dan sumbu y (vertikal). Dimana sumbu x akan merepresentasikan waktu, bisa dalam hari, bulan, atau tahun.sedangkan sumbu y akan merepresentasikan kuantitas. Jika *line* yang terdapat di dalam grafik dua atau lebih maka akan bisa digunakan untuk perbandingan.