BAB 2

TINJAUAN PUSTAKA

2.1 Profil Rumah Sakit Dera As-Syifa

2.1.1 Sejarah Rumah Sakit Dera As-Syifa

Pembangunan Rumah Sakit Dera As-Syifa dimulai dengan peletakan batu pertama pada tanggal 1 September 2006 di atas tanah bekas persawahan oleh panitia pembangunan lembaga kesehatan kecamatan Banjarharjo, kabupaten Brebes. Rumah Sakit Dera As-Syifa Banjarharjo terletak di tengah kota kecamatan Banjarharjo, tepatnya di jalan Merdeka No 119 kecamatan Banjarharjo, kabupaten Brebes. Berdiri di atas lahan seluas ±12.290m^2, Rumah Sakit Dera As-Syifa berada di lokasi yang cukup strategis, dengan ditunjang oleh kemudahan akses transportasi dan tingginya mobilisasi penduduk dari kecamatan di sekitar Banjarharjo.

2.1.2 Visi, Misi serta Tujuan Rumah Sakit Dera As-Syifa

2.1.2.1 Visi Rumah Sakit Dera As-Syifa

Slogan Rumah Sakit Dera As-Syifa adalah ” Memberikan Pelayanan Kesehatan Terbaik, Berkualitas, Profesional Dan Penuh Kasih Sayang Dengan Harga Terjangkau ”

Dalam hal ini Rumah Sakit Dera As-Syifa diwajibakn untuk bisa memberikan pelayanan kesehatan kepada masyarakat dengan mengedepankan kualitas terbaik, bekerja secara professional dan melayani masyarakat dengan penuh kasih sayang, serta dengan harga terjangkau. Dengan Begitu diharapkan Rumah Sakit Dera As-Syifa ini dapat memberikan pelayanan kesehatan terbaik dengan ditangani oleh tenaga-tenaga medis profesional, khususnya yang masih bisa terjangkau bagi kalangan masyarakat tingkat menengah ke bawah. Sehingga Rumah Sakit Dera As-

Syifa diharapkan bisa menjadi sebuah Rumah Sakit yang bisa mengayomi seluruh lapisan masyarakat.

2.1.2.2 Misi Rumah Sakit Dera As-Syifa

Adapun misi dari Rumah Sakit Dera As-Syifa berdasarkan Rencana Pembangunan Jangka Panjang Rumah Sakit Dera As-Syifa.

1. Meningkatkan pelayanan kesehatan yang bermutu dan terjangkau;
2. Memberikan pelayanan kesehatan terpadu sesuai kebutuhan pasien dan keluarga;
3. Melaksanakan pekerjaan dalam tim yang profesional, sinergi, inovatif, berdedikasi tinggi dan terpercaya;
4. Meningkatkan kualitas dan kuantitas sarana/prasarana pelayanan di semua bidang secara terus menerus berkesinambungan;
5. Mengembangkan SDM yang sehat, cerdas, berakhlak, professional dan berdaya saing;
6. Mengembangkan perekonomian, dalam hal ini untuk wilayah kecamatan Banjarharjo yang berdaya saing dalam menunjang menciptakan lapangan kerja dan pelayanan publik dalam pembangunan ekonomi;
7. Menciptakan lingkungan kerja yang sehat dan harmonis

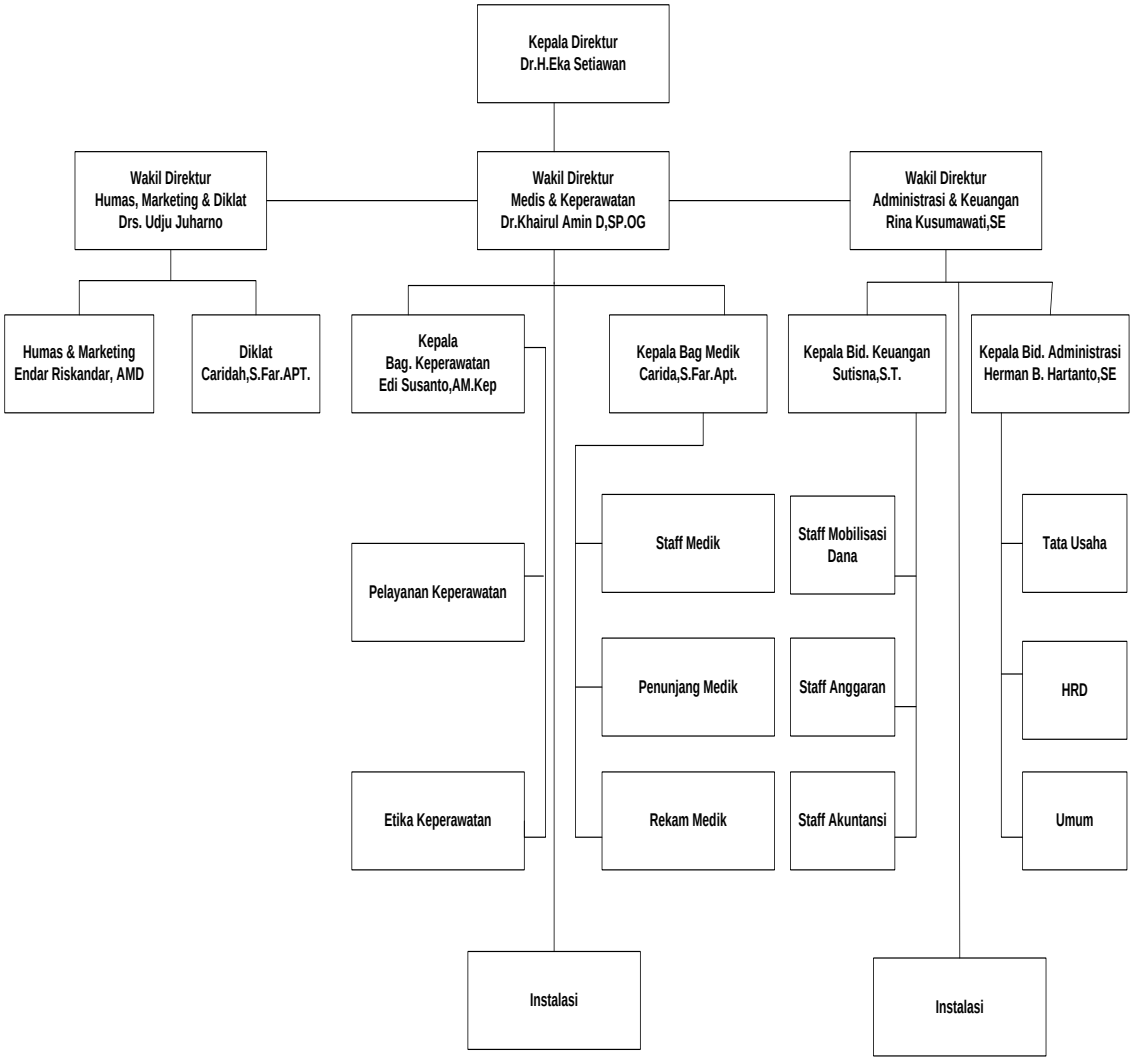
2.1.2.3 Tujuan Rumah Sakit Dera As-Syifa Banjarharjo

Tujuan berdirinya Rumah Sakit Dera As-Syifa adalah sebagai berikut:

1. Terwujudnya pelayanan kesehatan yang professional dan terbaik;
2. Terwujudnya pelayanan kesehatan dengan upaya optimalisasi derajat kesehatan masyarakat;

2.1.3 Struktur Organisasi Rumah Sakit Dera As-Syifa

Organisasi yang berlaku di lingkungan Rumah Sakit Dera As-Syifa kecamatan Banjarharjo adalah struktur organisasi yang berdaarkan pada aktifitas manajemen yang berjalan di dalam lingkungan Rumah Sakit Dera As-Syifa. Berikut adalah struktur organisasi dan susunan manajemennya.



Gambar 2.1 Struktur Organisasi Rumah Sakit Dera As-Syifa

Berikut *job description* dari Rumah Sakit Dera As-Syifa yang terlibat dengan pembangunan perangkat lunak *data mart.*

A. Kepala Bagian Medik

Kepala bagian medik merupakan pihak yang akan menggunakan perangkat lunak *data mart* selaku kepala dari bagian pendaftaran pasien dan penjualan. Kepala bagian medik membutuhkan sebuah sistem untuk mendapatkan informasi strategi

terkait rencana pengembangan layanan kesehatan ke berbagai lokasi di kecamatan banjarharjo.

2.2 Landasan Teori

2.2.1 Pengertian Data

Data adalah sebuah rekaman dari fakta-fakta, konsep-konsep, atau instruksi-instruksi pada media penyimpanan untuk komunikasi, perolehan, dan pemrosesan dengan cara otomatis dan presentasi sebagai informasi yang dapat dimengerti oleh manusia (Inmon, 2005).

Data adalah kumpulan fakta dan gambaran secara umum tidak dapat digunakan karena ukuran yang besar dan belum diolah (McLeod dan Schell, 2007).

Dari dua teori tersebut dapat disimpulkan bahwa data adalah kumpulan fakta yang secara umum tidak dapat digunakan karena belum diolah yang terdapat pada media penyimpanan untuk melalui pemrosesan sehingga dapat menyajikan informasi yang dapat dimengerti oleh manusia.

2.2.2 Pengertian *Data Warehouse*

Data *warehouse* adalah sebuah basis data yang mewakili sejarah bisnis dari suatu perusahaan atau organisasi yang berisikan data historis (Jose‘ Ramalho, 2001). *Data warehouse* menyimpan data dari berbagai sumber dan sifatnya tidak membuat data baru, data yang disimpan di dalam data *warehouse* sering diolah sebelum disajikan untuk *end-user*, misalnya diringkas sebelum dicetak dalam bentuk laporan.

Data yang terdapat pada *data warehouse* diatur sedemikian rupa sehingga dapat mendukung analisa. OLAP, atau *On-line Analytical Processing* adalah teknologi yang mengolah informasi dari *data warehouse* dalam struktur multidimensi, menyediakan jawaban yang cepat untuk *query*(memanggil data) analisis yang kompleks. Tujuan dari OLAP adalah mengorganisir sejumlah data yang besar agar bisa dianalisis dan dievaluasi dengan cepat (Chuck Ballard, 1998).

Fungsi utama data *warehouse* adalah mengambil data, termasuk data eksternal; misalnya daftar kode pos dari kantor pos atau kode kepala telepon, untuk dikumpulkan, disiapkan termasuk merubah struktur dengan pembersihan dan integrasi. Data *warehouse* menyimpan dan menyediakan data untuk aplikasi yang sifatnya *query* atau membentuk laporan ikhtisar yang selanjutnya dimanfaatkan dalam analisis informasi dan sebagai pendukung pembuatan keputusan (*analytical application*).

2.2.3 Karakteristik *Data Warehouse*

Menurut (Inmon, 2005), data *warehouse* memiliki beberapa karakteristik yaitu:

1. Berorientasi Subjek

   Data *warehouse* tidak berorientasi pada proses yang mengeksekusi fungsi dan data lain sebagaimana halnya pada operasi basis data transaksional. Pada data *warehouse*, data dikumpulkan menurut subjek asalnya. Sebagai contoh data pasien pada data *warehouse* di integrasikan menurut suatu entitas data, dan tidak diklasifikasikan menjadi beberapa entitas, untuk pendaftaran, pembelian obat dan begitu juga yang lainnya

2. Terpadu

   Data dari berbagai aplikasi transaksional memiliki spesifik yang sama, yang mana hal ini diterapkan pada sistem basis data relasional (RDBMS). Dengan adanya relasi kesamaan, data *warehouse* memiliki kemampuan integratif (terpadu) sehingga dapat menggabungkan berbagai bentuk data dan menyajikannya ke dalam bentuk yang konsisten sehingga dapat membandingkan dan memadukan data melalui berbagai dimensi yang sesuai.

3. Historis

   Waktu merupakan tipe atau bagian data yang sangat penting dalam data *warehouse*. Berbagai keterangan waktu data disimpan oleh data *warehouse*, misalnya waktu transaksi, waktu pemasukan data, dan lain-lain. Dengan kata

lain, setiap baris data merupakan data historis sehingga harus spesifik terhadap waktu, dengan demikian dapat menjaga konsistensi historis data yang disimpan.

4. Konsisten / *Non-volatile*

Data warehouse tidak mudah berubah. Data yang ada dalam suatu *data warehouse* tidak mengalami proses penghapusan atau perubahan, terkecuali untuk perawatan dan koreksi terhadap kesalahan yang terjadi. Data hanya disimpan dalam data *warehouse* atau ditampilkan dari data *warehouse.*

2.2.4 Keuntungan *Data Warehouse*

Menurut Connolly dan Begg (2005, p1152), data *warehouse* dapat memberikan keuntungan bagi perusahaan, diantaranya:

a. Tingkat pengembalian investasi yang tinggi

   Sebuah organisasi harus menangani sumber daya dalam jumlah besar untuk memastikan implementasi data *warehouse* yang berhasil dan biayanya bias sangat bervariasi. Berdasarkan penelitian dari *International* dan *corporation* pada 1996 rata-rata tingkat pembelian investasi data *warehouse* dalam tiga tahun mencapai 40%.

b. Keuntungan kompetitif

   Pengembalian dari investasi yang tinggi bagi perusahaan yang mengimplementasikan data *warehouse* dengan berhasil akan memberikan mereka keuntungan kompetitif. Keuntungan tersebut didapat dengan mengizinkan pembuat keputusan mengakses data yang sebelumnya tidak tersedia, tidak diketahui dan informasi yang belum bermanfaat, contohnya trend dan permintaan.

c. Meningkatkan produktivitas para pembuat keputusan perusahaan

   Data *warehouse* meningkatkan produktivitas dari pembuat keputusan perusahaan dengan menciptakan *database* yang terintegrasi, konsisten, berorientasi subjek dan data historis. Dengan merubah data menjadi informasi yang berarti, data

warehouse memungkinkan manajer untuk melakukan analisis dengan lebih akurat dan konsisten.

2.2.5 Perbandingan *Data Warehouse* dan *OLTP*

Secara umum perbedaan antara data *warehouse* dan OLTP (Connolly dan Begg, 2005,hlm 153) adalah:

Tabel 2.1 Perbedaan antara *OLTP dan Data Warehouse*

OLTP	**Data *Warehouse***
Menyimpan data saat ini	Menyimpan data histori
Menyimpan data detail	Menyimpan *detailed, lightly,* dan *highly summarized* data
Datanya bersifat dinamik	Datanya bersifat statis
Prosesnya berulang kali	Prosesnya khusus dan tidak terstruktur
Transaksi tingkat tinggi	Transaksinya tingkat menengah hingga rendah
Pola pengunaannya dapat diprediksi	Pola penggunaannya tidak dapat
Berfokus pada proses transaksi	Berfokus pada analisis
Berorientasi aplikasi	Berorientasi subjek
Mendukung pengambilan keputusan harian	Mendukung pengambilan keputusan strategis
Melayani pengguna dalam jumlah besar	Melayani pengguna dalam jumlah kecil biasanya manajer

2.2.6 Pengertian *Data Mart*

Data mart struktur data per department yang berasal dari *data warehouse* dimana data di denormalisasi berdasarkan kebutuhan informasi tiap epartmen (Inmon, 2005)

Data mart adalah *data warehouse* kecil yang dirancang untuk unit bisni strategis atau department. Keuntungan dari *data mart* adalah biayanya rendah, waktu implementasinya singkat, pengendaliannya local bukan terpusat (Turban, 2005).

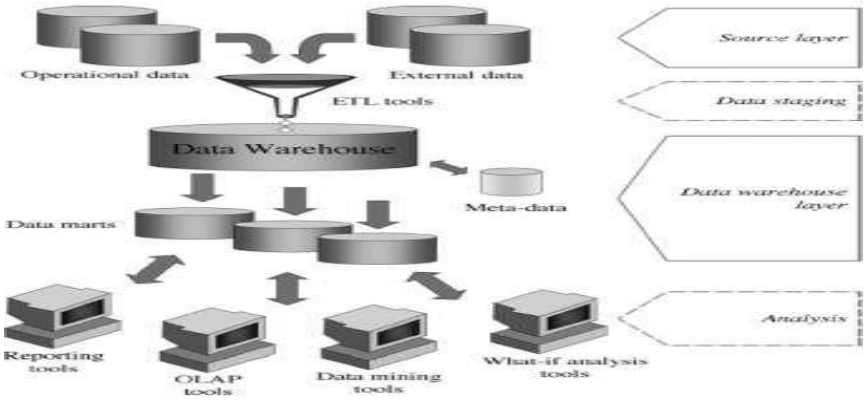
Data mart dan *data warehouse* memiliki perbedaan (Ponniah, 2001), yaitu

Tabel 2.2 Perbedaan antara *data Warehouse* dan *Data mart*

Data Warehouse	***Data Mart***
Merujuk pada keseluruhan organisasi	Merujuk pada department tertentu
Gabungan semua dari data mart	Sebuah proses bisnis
Data diterima dari staging area	Star join (fakta dan dimensi)
Di-Query di sumber tampilan	Teknologi yang maksimal untuk akses dan analisis data
Membentuk pandangan data untuk organisasi	Membentuk pandangan data per department
Diorganisasikan dalam ER-model	

2.2.7 *Two-Layer Architecture*

Adaa beberapa jenis arsitektur *data mart* yang dapat digunakan dalam pembangunan *data mart*. Salah satunya adalah *two-layer architecture*. Menurut dalam buku yang berjudul *Data Waraehouse Design : Modern Principles and Methodologies; (*Matteo Golfarelli and Stefano)



Gambar 2.2 *Two- Layer Architecture*

2.2.8 Jenis-jenis *data source* untuk *data mart*

Data source dapat dibedakan menurut beberapa jenis kebutuhanya, dan berdasarkan sumber data yang diperoleh. berikut ini adalah jenis-jenis dari *data source* :

1. *On-Line Transaction Processing (OLTP)* : Merupakan sebuah sistem yang memfasilitasi dan mengelola aplikasiyang berorentasi transaksi Biasanya untuk data entri ataupengambilan proses transaksi.
2. *Sistem Legacy* : Sistem yang sudah out-of-date tetapi data-datanya masih digunakan untuk keperluan historical.
3. *External Data* : Data yang berasal dari pihak di luar perusahaan.
4. *Middleware*: Sebuah aplikasi yang menjembatani antar aplikasi yang berbeda
5. *Source-less data* : Data yang sudah tidak ada lagi sumber datanya.

2.2.9 *Staging Area*

Proses ETL (*Extract-Transform-Load*) terjadi mulai dari ketika data memasuki hingga keluar dari *Staging* Area. Pada tahap ini, data diekstrak (Extract) 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 files*), 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.

a. Extraction

Data Extraction adalah proses pengambilan data yang diperlukan dari sumber *data mart* dan selanjutnya dimasukkan pada staging area untuk diproses pada tahap berikutnya . Pada fungsi ini banyak berhubungan dengan berbagai tipe sumber data seperti: Format data, mesin yang berbeda, software dan arsitektur yang tidak sama. Sehingga sebelum proses ini dilakukan, sebaiknya perlu

didefinisikan requirement terhadap sumber data yang yang akan digunakan untuk proses berikutnya.

b. Transformation

Pada kenyataannya, pada proses transaksional data disimpan dalam berbagai format sehingga jarang kita temui data yang konsisten antara aplikasi-aplikasi yang ada. Transformasi data ditujukan untuk mengatasi masalah ini. Dengan proses transformasi data ini, kita melakukan standarisasi terhadap data pada satu format yang konsisten. Beberapa contoh ketidakkonsistenan data tersebut dapat diakibatkan oleh tipe data yang berbeda, data length dan lain sebagainya.

c. Load

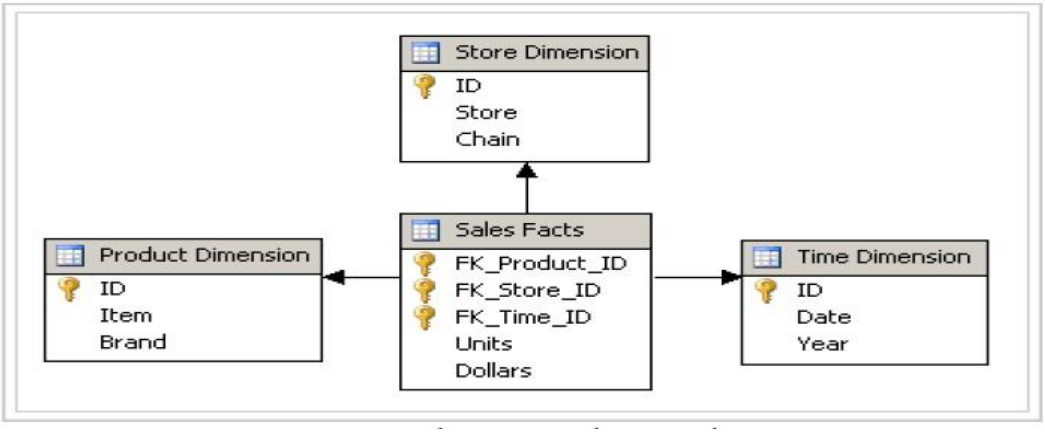
Data load adalah memindahkan data ke *data mart*. Ada dua loading data yang dilakukan pada *data mart*. Pertama adalah inisial load, proses ini dilakukan pada saat telah selesai mendisain dan membangun *data mart*. Data yang dimasukkan akan sangat besar dan memakan waktu yang relati lebih lama. Kedua Incremental load, dilakukan ketika *data mart* telah dioperasikan. Incremental load ini dapat dilakukan sesuai dengan sistem yang dibangun.

2.2.10 Skema *Data Mart*

Ada beberapa skema yang terdapat didalam pemodelan *data mart*, seperti skema star atau skema snowflake. Keduanya merupakan moel-model dimensional, perbedaanya terletak pada implementasi fisikal. Skema snowflake memberi kemudahan pada perawatan dimensi, dikarenakan strukturnya yang lebih normalisasi. Sedangkan skema bintang lebih efisien serta sederhana dalam membuat query dan mudah di akses secara langsung oleh pengguna.

a. Skema Bintang

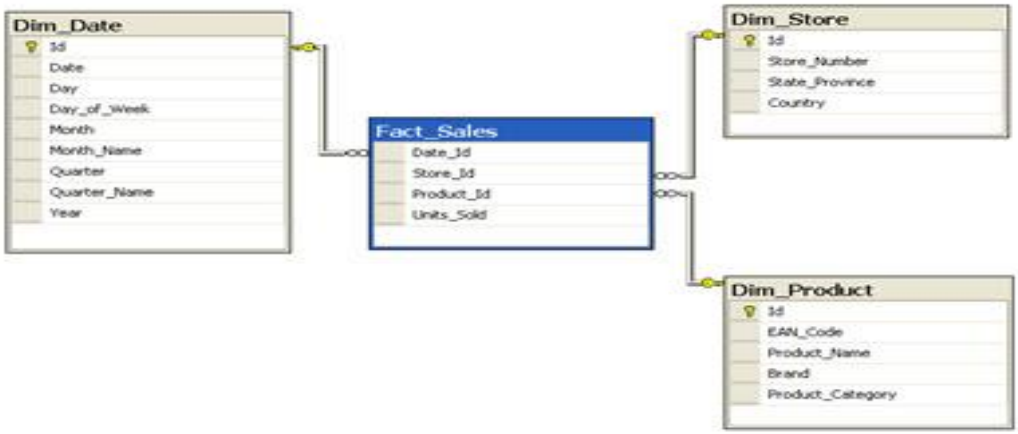
Skema bintang adalah sebuah logikal struktur yang mempunyai sebuah tabel fakta berisi data terbaru di tengah, yang dikelilingi tabel dimensi yang berisi data referensi (yang dapat didenormalisasi), (Connolly dan Begg, 2005)



Gambar 2.3 Skema Bintang

b. Skemas Snowflake

Merupakan varian dari skema bintang dimana table-table dimensi tidak terdapat data yang di denormalisasi. Dengan kata lain satu atau lebih table dimensi tidak bergabung secara langsung kepada table fakta tapi pada table dimensi lainnya.



Gambar 2.4 Skema Snowflake

2.2.11 *Online Analytical Processing* (OLAP)

On-line Analytical Processing, atau disingkat OLAP adalah teknologi yang memproses data *warehouse* dalam struktur multidimensi, menyediakan jawaban yang cepat untuk *query* analisis yang kompleks. Tujuan dari OLAP adalah mengorganisir sejumlah data yang besar, agar bisa dianalisa dan dievaluasi dengan cepat dengan

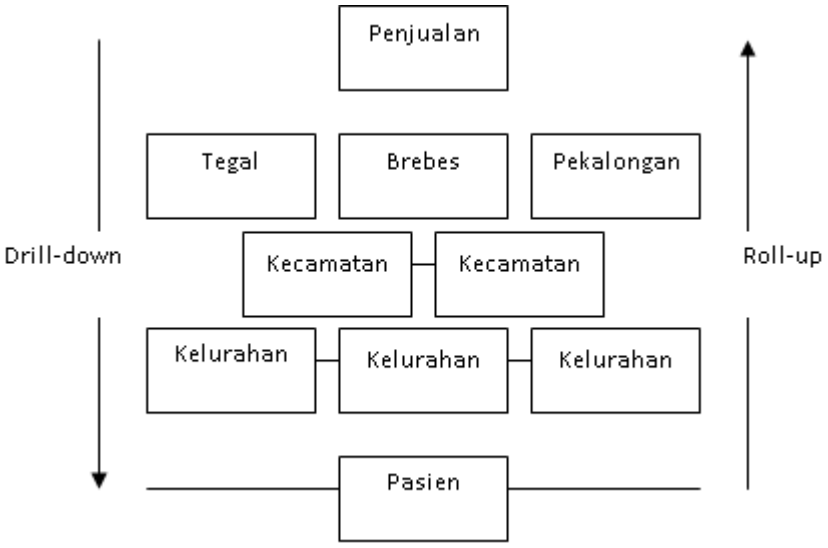
menggunakan sarana *online* dan representasi grafik. *Query* untuk OLAP biasanya lebih sulit dari OLTP karena bersifat pengganti dan tidak melibatkan operasi data update. OLAP tidak memakai data operasi sehari-hari begitu saja, tetapi memakai data yang sudah terangkum dengan model data yang disebut *data cube.* OLAP mengorganisir data dalam *data mart* ke struktur kubus multidimensi berdasarkan model dimensional yang diproses dengan *query* analisis.

2.2.12 Multidimensi

Analisa secara multidimensi menjadi cara yang semakin populer untuk memperluas kemampuan yang dimiliki dari pengolahan *data mart* melalui *query and reporting*. Dalam analisa multidimensional, data terlebih dahulu dibuatkan struktur agar dapat memberi kecepatan dan kemudahan dalam mengakses informasi dalam menjawab pertanyaan-pertanyaan yang biasa dipertanyakan, hal ini dianggap lebih baik daripada menggunakan banyak *query*. Sebagai contoh, data akan distruktur untuk menjawab pertanyaan, "Berapa banyak pengeluaran harian pada setiap bagian/divisi?" Setiap bagian dari *query* dinamakan sebuah *dimension* (dimensi). Dengan menyiapkan terlebih dahulu jawaban untuk setiap *query* dengan konteks yang lebih besar, maka banyak jawaban yang siap digunakan karena hasilnya tidak perlu dihitung lagi dengan masing-masing menggunakan *query*. Dimensi-dimensi dapat mempunyai entitas individual atau hierarkis; seperti wilayah dan departemen/divisi.

Analisa secara multidimensi memungkinkan para pengguna untuk melihat pada sejumlah besar faktor-faktor yang saling berhubungan dalam suatu perihal bisnis dan untuk melihat data dalam keterkaitan yang lebih kompleks. *End-user* akan lebih tertarik untuk melakukan *exploring* pada data dengan berbagai level secara detail, yang ditentukan secara dinamik. Keterkaitan yang kompleks dapat dianalisa melalui suatu proses yang terdiri dari *drilling down* atau *rolling up* dari berbagai level peringkasan dan pengelompokan. *Drilling down* mengelompokkan data dari level tinggi ke level rendah, misalnya pengelompokan data berdasarkan alamat pasien dimulai dari kabupaten, kecamatan, dan kelurahan. Sedangkan *rolling up*

mengelompokkan data dari level yang rendah ke level yang tinggi misalnya pengelompokan alamat costumer dimulai dari kelurahan, kecamatan, dan kabupaten. Proses *drilling down* dan *rolling up* dapat dilihat pada gambar 2.6.



Gambar 2.5 Analisis *Drill-down* dan *Roll-up (Chuck Ballard, 1998)*

2.2.13 Multidimensional Sebagai Model *Data Mart*

Multidimensional model adalah teknik untuk memodelkan bisnis secara konseptual sebagai sekumpulan ukuran yang dijabarkan oleh segi bisnis secara umum. Sangat berguna untuk pergantian, peringkasan dan penyusunan data untuk analisis. Sangat kontras dengan sistem OLTP yang didesain di sekitar entitas, *relationship, functional decomposition*, dan state *transition analysis. Multidimensional model* untuk data *warehousing* dibuat berdasarkan pada fakta, dimensi, hierarki dan *sparsity*.

Multidimensional model sangat berguna untuk meringkas dan menyusun data dan memperlihatkan data untuk mendukung para analis data. *Multidimensional model* memfokuskan pada data numerik, seperti harga, jumlah, berat dan lain-lain. Penerapan pemodelan multidimensi dapat mengorganisir data dari *data mart* di dalam kubus-kubus multidimensi dengan informasi rangkuman yang telah dihitung sebelumnya yang disebut dengan *data cube*, dimensi informasi yang dihasilkan akan

berfungsi untuk memberikan jawaban yang cepat untuk *query* analisis yang kompleks. *Data cube* adalah presentasi data multidimensi seperti jenis barang, waktu, lokasi dan nilai harga.

Data cube yang tersedia pada *data mart* memungkinkan pemakai untuk menganalisa data operasi sehari-hari, bulanan hingga tahunan dengan berbagai sudut pandang yang berbeda, dan sangat berguna untuk mengevaluasi suatu asumsi bisnis.

Bagian yang menarik dari Multidimensional Model adalah kesederhanaannya. Desain OLTP boleh jadi memiliki selusin atau bahkan ribuan tabel, yang menyebabkan kesulitan bagi manajer untuk memahami desain data dalam rangka menganalisis data. Multidemensional Model lebih sederhana, model tidak hanya dimengerti oleh orang bisnis, tetapi juga diekspresikan dengan cara yang natural oleh *user*. Desain konseptual Multidemensional Model yang baik, dapat diimplementasikan dalam *database* relasional, multidimensional, bahkan *object-oriented database*. Keuntungan mengelola data menggunakan Multidimensional modelling (MDM) adalah:

a. Mudah digunakan karena menyediakan antarmuka *user*.
b. Model data yang fleksibel dan tangguh untuk definisi dan penyimpan kubus.
c. Solusi yang terotomatisasi untuk mengatasi ledakan data.
d. Arsitektur yang memiliki skalabilitas yang tinggi, mampu mendukung beberapa server data *warehouse* OLAP.
e. Memberikan jawaban yang cepat atas *query* dan mengurangi lalu-lintas *network*.

Multidimensional modelling mempunyai beberapa konsep yaitu: *Facts, Dimensions*, dan *Measures* yang akan dijelaskan pada bagian berikut ini:

A. *Fact*

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

proses bisnis. Dalam *data mart, fact* diimplementasikan dalam tabel dasar dimana semua data numerik disimpan.

B. *Dimensons*

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 dampak dari dimensi pada *facts.*

Dimensi adalah parameter yang diinginkan dalam OLAP. Sebagai contoh, dalam suatu *database* untuk menganalisa semua kegiatan penjualan dari produk, dimensi yang harus ada yaitu waktu, lokasi, pembeli, penjual.

Ada beberapa jenis *dimensions* dalam data *warehouse*:

a. *Dimension members*

Dimension members adalah suatu dimensi berisi banyak anggota-anggota member. Suatu anggota dimensi adalah nama pembeda atau *identifier* yang digunakan untuk membedakan posisi suatu data item. Sebagai contoh, bulan, kuartal dan tahunan membuat dimensi waktu, sedangkan kota, wilayah dan negara merupakan dimensi geografi.

b. *Dimension Hierarchies*

Dimension Hierarchies adalah suatu dimensi yang berfungsi mengatur anggota dari suatu dimensi ke dalam satu atau banyak hierarki. Setiap hierarki dapat juga mempunyai banyak level hierarki. Setiap member dari dimensi tidak dialokasikan ke dalam satu struktur hierarki.

C. *Measures*

Suatu *measures* (ukuran) adalah suatu besaran (angka numerik) sebagai atribut dari sebuah *fact*, yang menunjukkan *performance* atau *behaviour* (tingkah laku) dari bisnis secara relatif pada suatu dimensi. Angka atau nomor yang ditunjukkan disebut

dengan variabel. Sebagai contoh ukuran dari penjualan adalah dalam bentuk uang, besarnya penjualan, jumlah persediaan, biaya belanja, jumlah transaksi dan lainnya. Suatu ukuran dijelaskan dengan kombinasi dari member dari suatu dimensi dan diletakkan dalam *fact*.

2.2.14 *Cube*

Cube adalah rangkuman informasi yang disimpan dalam bentuk tabel-tabel multi dimensional. Dalam *data mart* terdapat tiga jenis tampilan *cube* yang berbeda yaitu:

a. Panel *tree view* yang berfungsi menampilkan dimensi kubus secara hirarki.
b. Panel *scheme* berfungsi menampilkan tabel-tabel yang sesungguhnya dari *data mart*, tabel dimensi, serta hubungan antar tabel yang mendefinisikan skema *cube*.
c. Panel *Browse* berfungsi menampilkan data yang ada di dalam kubus.

Rangkuman informasi atau *cube* yang telah dibuat dapat disimpan dengan metode penyimpanan yang beragam diantaranya:

A. MOLAP

Multidimensional OLAP, data dari kubus disimpan bersama-sama dengan data koleksi di dalam struktur multidemensi.

B. ROLAP

Relational OLAP, data dasar kubus disimpan bersama data koleksi di dalam *database* relasional. Penyimpanan dengan ROLAP ini mengizinkan user memaksimalkan investasi pada teknologi *database* relasional.

C. HOLAP

Hybrid OLAP, data dasar kubus disimpan di dalam *database* relasional, data koleksi disimpan di dalam struktur multidimensi dengan kinerja tinggi. Penyimpanan dengan HOLAP bisa menyimpan data seperti yang ada pada MOLAP tanpa membutuhkan data dasar yang detail.

Data dalam kubus merupakan data fleksibel karena:

1. Penyimpan berpartisi

   User bisa mempartisi sebuah kubus logik dalam bagian-bagian yang dipisahkan secara fisik. Dan hasil dari partisi bisa disimpan dalam tampilan yang berbeda, dan dengan tingkat koleksi yang sesuai dengan data yang dipartisi. Dengan adanya penyimpanan berpartisi ini *user* bisa mengatur kinerja dan karakteristik dari manajemen data dalam sistem yang dibuat.

2. Menggabungkan partisi

   Kubus yang sudah dipartisi dapat digabungkan kembali partisi-partisinya ke dalam bentuk satu kubus fisik. Salah satu kegunaan penggabungan partisi adalah untuk memindahkan bagian dari data kubus ke media penyimpanan yang berbeda, sehingga data lebih sedikit digunakan.

3. Kubus virtual

   Kubus-kubus fisik bisa digabungkan ke kubus virtual, sama seperti tabel digabungkan ke dalam *view* di dalam *database* relasional. Kubus virtual menyediakan akses terhadap data-data yang ada pada kubus yang digabungkan, tanpa perlu membuat kubus baru.

2.2.15 Alat Pembantu Dalam Perancangan Sistem

Perancangan *system* secara umum adalah suatu tahap dimana di dalamnya terdapat identifikasi komponen-komponen system infomasi yang akan dirancang

secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau user mengenai *system* yang baru. Sedangkan desain *system* secara terinci dimaksudkan untuk pembuat program komputer dan ahli teknik lainnnya yang akan mengimplementasikan sistem.

Penggambaran dan rancangan model *system* infomasi secara logika dapat dibuat dalam bentuk diagaram konteks dan Diagram Alir Data (DAD) atau *Data Flow Diagram (DFD).*

2.2.15.1 Diagram Konteks

Diagram kontek merupakan alat-alat untuk struktur analisis. Pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan. Diagram kontek adalah kasus khusus dari DFD atau bagan dari DFD

yang berfungsi memetakan modul lingkungan yang dipersentasika dengan lingkaran tunggal yang mewakili keseluruhan sistem.

2.2.15.2 *Data Flow Diagram* (DFD)

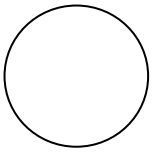
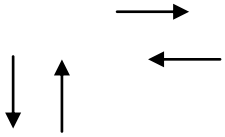
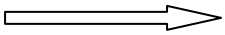
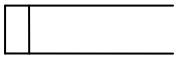
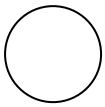
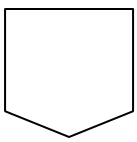
Diagram alir data atau *Data Flow Diagram (DFD)* adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagaram konteks yang menjelaskan cecara umum suatu *system* atau batasan system aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi menjadi level 1 dan selanjutnya sampai system tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut:

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2. Penurunan dilakukan apabila memang diperlukan.
3. Tidak semua bagian dari system harus ditunjukkan dengan jumlah level yang sama.

Symbol-simbol yang digunakan pada diagram alir data, atau Data Flow Diagram antara lain:

Tabel 2.3 Daftar Simbol dalam *Data Flow Diagram* (DFD)

Simbol	Pengertian	Keterangan
	Sistem	Menunjukan sistem
	Eksternal *entity*	Menunjukan bagian luar sistem atau sumber input dan output data
	Garis aliran	Menunjukan arus data antar simbol/proses
	Garis aliran	Aliran material
	Data *Storage*	Digunakan untuk menyimpan arus data atau arsip seperti file transaksi, file induk atau file referensi dan lain-lain
	Proses	Suatu proses yang dipicu atau didukung oleh data.
	Conector (*On-page connector*)	Digunakan untuk penghubung dalam satu halaman
	Conector (*Off-page connector*)	Digunakan untuk penghubung berbeda halaman

2.2.16 Pengertian Basis Data

Menurut Caonnolly dan Begg (2005, p15), *database* adalah kumpulan dari data yang saling berhubungan secara logikal, dan menjelaskan data tersebut, dirancang untuk memenuhi kebutuhan informasi perusahaan.

Menurut Inmon (2005, p493), *database* adalah kumpulan data yang saling berhubungan yang disimpan berdasarkan skema.

Menurut McLeod dan Schell (2007, p124) *database* adalah kumpulan dari semua data berbasis komputer perusahaan.

Dapat disimpulkan bahwa *database* adalah kumpulan dari data yang dimiliki perusahaan yang berhubungan secara logikal dan disimpan berdasarkan skema yang dirancang untuk memenuhi kebutuhan informasi perusahaan

2.2.17 Konsep Dasar Basis Data

Basis data (*database*) dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis data menunjukkan suatu kumpulan data yang dipakai dalam system informasi disebut sistem basis data(*database system*)

2.2.17.1 Basis Data Relasional

Konsep sebuah *database* adalah terdiri atas tabel-tabel yang terorganisasi. Tabel-tabel tersebut dapat saling berelasi untuk menghasilkan suatu infomasi, untuk mengakses data yang ada dalam tabel-tabel tersebut digunakan sebuah perintah SQL*(Structured Query language).*

2.2.17.2 Perancangan Basis Data

Perancangan basis data dibutuhkan agar diperoleh suatu sistem yang sesuai dengan apa yang diinginkan, dimana dapat melalui tahapan berikut :

1. ERD *(Entitiy Relationship Diagram)*

Entity Relationship Diagram adalah suatu model atau teknik pendekatan yang dapat menyatakan atau menggambarkan hubungan *entity* didalam sebuah sistem, dimana hubungan tersebut dinyatakan sebagai *one-to-one, one-to-many* dan *many-tomany*. *Entity relationship* bertujuan untuk menentukan objek-objek data (*Entity*) dan *relationship* (hubungan) yang ada pada objek-objek tersebut.

Berikut adalah istilah-istilah yang sering digunakan dalam perancangan *entity relationship*:

a. Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain.
b. Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.
c. Kardinalitas/derajat relasi menunjukkan jumlah maksimum entitas yang dapat berhubungan dengan entitas pada himpunan entitas yang lain.

Entity-Relationship Diagram (ERD) didefinisikan sebagai penggambaran model ER yang berisi komponen-komponen himpunan relasi beserta atribut-atributnya secara sistematis (Fathansyah2002). Notasi-notasi simbolik di dalam ERD yang digunakan antara lain sebagai berikut:

a. Persegi panjang menyatakan himpunan entitas
b. Lingkaran/Elips menyatakan atribut(atribut yang berfungsi sebagai *primary key* diberi garis bawah)
c. Belah ketupat menyatakan himpunan relasi
d. Garis sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya
e. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk menyatakan relasi satu ke satu, 1 dan N untuk menyatakan relasi satu ke banyak dan N dan N untuk menyatakan relasi banyak ke banyak)

2. Perancangan Model Logis

Model Logis merupakan model ER yang sudah dinormalisasi. Normalisasi di sini maksudnya adalah setiap entitas yang masih memiliki hubungan banyak ke banyak dengan entitas lainnya maka hubungannya tersebut akan dipecah menjadi suatu entitas baru. Model ini juga sudah dilengkapi dengan tipe data yang bersifat generik. Model logis merefleksikan metadata yang sudah siap diubah menjadi model fisik. Model ini umum direpresentasikan sebagai suatu tabel.

3. Perancangan Model Fisik

Model fisik merupakan model yang menunjukkan implementasi model logis di suatu DBMS tertentu. Model ini sudah memiliki tipe data dan konstrain yang bersifat spesifik terhadap target DBMS-nya. Model fisik merefleksikan metadata yang sudah diimplementasikan di atas sebuah DBMS tertentu, seperti *MySQL Server*.

2.2.18 *Microsoft Visual FoxPro 7*

Microsoft visual FoxPro merupakan bagian dari paket aplikasi Microsoft Visual Studio. Visual FoxPro 7 menerapkan sistem manajemen basis data relasional (RDBMS) berorientasi objek yang memiliki kemampuan untuk membuat aplikasi database dengan cepat dan memiliki fasilitas yang lengkap serta mudah digunakan. Bahasa program ini pun memungkinkan pengembangan aplikasi database untuk desktop dalam suatu solusi *client-server*.

2.2.18.1 Komponen *Visual FoxPro*

Dalam Visual FoxPro, dikenal beberapa istilah dan komponen yang sering dipakai dalam membuat aplikasi. Beberapa komponen tersebut yang akan dibahas yakni, Project Manager, Form Designer, Properties, dan Form Control.

a. Project Manager

Project Manager adalah sekumpulan Form, Database, Tabel, Menu, Class Library dan lainnya. Secara singkat, Project Manager berguna untuk menata dan mengontrol sejumlah *tool* yang akan digunakan dalam membangun aplikasi.

b. Form Designer

Form ialah suatu objek yang digunakan sebagai tempat kerja aplikasi. Form diibaratkan sebagai lembar kerja dimana setiap objek dapat ditempatkan di atasnya dalam membentuk antarmuka (*interface*) aplikasi. Form Designer berfungsi sebagai tempat untuk menciptakan antarmuka aplikasi.

c. Properties

Properties digunakan untuk menentukan aturan objek. Suatu objek pada umumnya memiliki sejumlah sifat dan *events* tersendiri, yang masing-masing dapat diatur langsung pada *Properties window* ini atau melalui kode program. Pengaturan sifat dan *events* setiap objek akan mempengaruhi cara kerja objek tersebut.

d. Form Control

Form Control merupakan kumpulan objek yang dapat ditempatkan di atas *Form* dan digunakan dalam membangun aplikasi. Objek-objek tersebut berupa *CommandButton, Textbox, Label* dan lain-lain.

2.2.19. MySql

MySQL adalah sebuah perangkat lunak yang digunakan untuk mengolah database pada konsep *client-server*, dimana MySQL ini didistribusikan secara gratis. Dengan menggunakan *database* ini, data semakin aman dan berdaya guna. *Database* ini juga banyak dipakai pada web *database* sehingga data semakin terintegrasi antara *database* dekstop dengan *database* web. Untuk menggunakan *database* MySQL harus menginstalasinya dahulu ke komputer.

Keistimewaan MySQL Sebagai server database dengan konsep database modern, MySQL memiliki keistimewaan. Beberapa keistimewaan dimiliki MySQL sebagai berikut :

1. *Portability*

   Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-Unix, dan lain-lain.

2. *Open Source*

   MySQL merupakan *database open source* (gratis), di bawah lisensi GPL sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.

3. *Multiuser*

   MySQL merupakan *database* yang dapat digunakan untuk menangani beberapa user dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah *database* server MySQL dapat diakses client secara bersamaan pula.

4. *Performace Tuning*

   MySQL mempunyai kecepatan yang cukup baik dalam menangani query query sederhana, serta mampu memproses lebih banyak SQL per satuan waktu.

5. *Column Type*

   Database MySQL didukung dengan tipe data yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set* serta *enum.*

6. *Command dan Functions*

   MySQL server memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

7. *Security*

   Sistem Security pada MySQL mempunyai beberapa lapisan sekuritas seperti tingkatan *subnetmask, hostname,* dan izin akses user dengan sistem perizinan yang mendetil serta *password* terenkripsi.

8. *Scalability dan Limits*

   MySQL mempunyai kemampuan menangani *database* dalam skala cukup besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu dapat menampung indeks sampai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   Adanya kemampuan MySQL melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).

10. *Localization*

    Adanya kemampuan dalam mendeteksi kesalahan (error code) pada client menggunakan lebih dari dua puluh bahasa.

11. *Interface*

    MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemograman menggunakan fungsi API (Application Programming Interface).

12. *Clients dan Tools*

    Database MySQL dilengkapi berbagai tools yang dapat digunakan untuk administrasi *database.*

13. Struktur Tabel

    MySQL memiliki struktur tabel cukup baik serta cukup fleksibel, misalnya ketika menangani alter table

2.2.20. Arsitektur *Client-Server*

Ada beberapa vendor yang menyediakan fasilitas *client server* salah satunya yaitu *Microsoft Decision Support Service*. *Microsoft Decision Support Service* mampu menyediakan kapasitas server untuk pembuatan dan pengelolaan data OLAP multidemensi serta untuk menyediakan data kepada client melalui *Microsoft Pivot table Service*.

a. Arsitektur *server*

Operasi pada server mencakup pembuatan kubus data multidimensi dari *database* relasional yang terdapat pada data *warehouse*, serta penyimpanan kubus-kubus di dalam struktur multidimensi, pada *database* relasional, dan gabungan dari kedua-duanya. Metadata dari struktur multidimensi dari sebuah kubus disimpan dalam penyimpanan *database* relasional. Yang dimaksud dengan metadata yaitu bagian dari data yang sangat menentukan agar data yang dimiliki atau data yang dikelola menghasilkan informasi yang benar dan berguna untuk para pengambil keputusan.

b. Arsitektur *Client*

Fungsi *client* dalam data *warehouse* adalah tempat menyimpan file-file multidimensi sama seperti kubus ROLAP pada *database* relasional dan mengakses data OLAP di dalam server. Kubus lokal bisa digunakan secara *ofline* dengan *PivotTable Service* untuk analisa yang portabel. *PivotTable Service* berkomunikasi dengan server melalui protokol pribadi.