BAB 2
TINJAUAN PUSTAKA

2.1 Profil Institusi

PT Pupuk Iskandar Muda atau biasa disebut dengan PT PIM adalah anak perusahaan PT Pupuk Indonesia (Persero) yang bergerak di bidang isdustri kimia khususnya memeproduksi pupuk urea. PT PIM merupakan pabrik pupuk skala besar pertama yang dibangun oleh putra putri bangsa melalui kontraktor nasional PT Rekayasa Industri pada tahun 1995. PT PIM berdiri dengan strategi untuk mencukupi kebutuhan pupuk urea dikawasan indonesia bagian barat yang secara geografis termasuk kawasan pertanianan, setelah sebelumnya kebutuhannya dirintis PT Pusri Palembang.

2.2 Logo

Berikut ini adalah logo dari PT Pupuk Iskandar Muda yang dapat dilihat pada gambar 2.1



Gambar 2.1 Logo PT. Pupuk Iskandar Muda

2.3 Visi dan Misi

2.3.1 Visi

Visi dari PT Pupuk Iskandar Muda adalah menjadi perusahaan pupuk dan petrokimia yang kompetitif.

2.3.2 Misi

Misi dari PT Pupuk Iskandar Muda adalah sebagai berikut :

1. Memproduksi dan memasarkan pupuk dan produk petrokimia dengan efisien.
2. Memenuhi standar mutu dan berwawasan lingkungan.
3. Memberikan nilai tambah kepada stakeholder.
4. Berperan aktif menunjang ketahanan pangan.

2.4 Landasan Teori

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

2.5 Data Warehouse

Data *warehouse* adalah sebuah *database* yang mewakili sejarah bisnis suatu perusahaan atau organisasi. Data *warehouse* merupakan kombinasi dari *database* yang berbeda-beda. Data *warehouse* mengkombinasikan informasi dengan meringkas (*summarizing*) dan mengelompokkan (*agregation*). Informasi yang diperlukan ini didefinisikan oleh 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. [2]

Data *warehouse* adalah suatu sistem komputer untuk mengarsipkan dan menganalisa data historis suatu organisasi seperti data penjualan, gaji, dan informasi. Menurut W.H. Inmon dan Richard D.H., data *warehouse* adalah

koleksi data yang mempunyai sifat beriorentasi subjek, terintegrasi, dan bersifat tetap dari koleksi data dalam mendukung proses manajemen pengambilan keputusan. Menurut Vidette Poe, data *warehouse* merupakan basis data yang bersifat analisis dan *read only* yang digunakan sebagai pondasi dari sistem penunjang keputusan. Menurut Paul Lane, data *warehouse* merupakan basis data relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, biasanya mengandung data historis dari proses transaksi dan bisa juga data dari sumber lainnya. Data *warehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung data dari berbagai macam sumber.

Data *warehouse* adalah multidimensional dalam relasional model *database*, informasi ditentukan dalam suatu rangkaian dari tabel yang berdimensi dua. Tidak begitu halnya dengan data *warehouse*, banyak data *warehouse* adalah multidimensional, artinya bahwa data *warehosue* terdiri dari *layer-layer*, kolom dua baris. *Layer-layer* dalam suatu data *warehouse* menunjukkan informasi disampaikan pada dimensi yang berbeda. Multidimensional ini menggambarkan informasi dengan ditunjukkan sebagai suatu kubus. Data *warehouse* adalah bentuk khusus dari *database*. Mengingat hal bahwa suatu *database* adalah koleksi dari informasi yang dikumpulkan dan diakses melalui suatu logikal dari informasi, begitu juga hal yang sama untuk suatu data *warehouse*.

Data *warehouse* mendukung proses pembuatan keputusan, bukan proses transaksi. Dalam suatu organisasi, banyak *databse* yang ada merupakan *database* yang berorientasikan (OLTP) oleh karena itu merupakan operasional *database*. Data *warehouse* ada untuk mendukung berbagai macam pekerjaan pengambilan keputusan dalam suatu organisasi. Definisi klasik oleh Bill Inmom tentang data *warehouse* dapat disimpulkan menjadi empat kriteria[2] :

1. Sebuah data *warehouse* adalah *subject-oriented* (beriorentasi pada subjek). Data suatu organisasi berubah dari yang beriorentasi pada aplikasi ke beriorentasi pada subjek. Menurut hal ini, daripada mengorganisasikan data dengan penyesuaian pada aplikasi atau fungsi-fungsi yang ada, data lebih baik dikumpulkan berdasarkan subjek area asalnya. Misalkan data tentang konsumen

yang diintegrasikan pada sebuah entitas *database*, akan lebih baik daripada mengklasifikasikannya menjadi beberapa entitas untuk proses peminjaman, begitu juga lainnya.

2. Data *warehouse* adalah terintegarasi yang menggabungkan berbagai format data dan menyediakannya kedalam suatu bentuk yang konsisten sehingga dalam membandingkan dan mengumpulkan data dengan melalui berbagai dimensi yang sesuai.

3. Data *warehosue* adalah *time-variant*. Dalam kata lain, setiap *rows* dari data dibedakan berdasarkan waktu. Dalam hal ini setiap baris biasanya memiliki satu atau lebih yang menggunakan tipe *data time-stamp*.

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

Keempat karakteristik di atas saling terkait dan harus diimplemetnasikan agar terbentuk suatu data *warehouse* yang dapat mendukung pengambilan keputusan secara efektif. Implementai dari keempat karakteristik di atas membutuhkan struktur data dari data *warehousing,* data dari berbagai sumber operasional akan diekstrak dan diintegrasikan ke dalam data *warehousei* sehingga data yang dihasilkan tidak lagi bersifat operasional melainkan informatif.

2.5.1 **Tugas Data** Warehouse

Ada tiga tugas yang bisa dilakukan dengan adanya data *warehouse*, keempat tugas tersebut yaitu [3]:

a. Pembuatan laporan

Pembuatan laporan merupakan salah satu kegunaan data *warehouse* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan per hari, per bulan, per tahun atau jangkan waktu kapanpun yang diinginkan.

b. *Online Analytical Processing* (OLAP)

Dengan adanya data *warehouse*, semua informasi baik *detail* maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah didapat. OLAP mendayagunakan konsep data multidimensi dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikan satupun perintah SQL. Hal ini dimungkinkan karena pada kosnep multidimensi, maka data yang berupa fakta yang sama bisa dilihat dengan menggunakan fugnsi yang berbeda.

c. Proses informasi executive

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

2.5.2 Proses Dan Arsitektur Data *Warehouse*

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*).

Sesuai dengan namanya, aplikasi ETL melakukan fungsi-funsi *extract, transform* dan *load*. Proses *extract* adalah proses pengambilan data dari sumber data. Disebut *extract*, karena proses pengambilan data ini tidak mengambil data matang aja. Proses *extract* ini harus mengakomodir berbagai macam teknologi yang digunakan oleh sumber data dan diintegrasikan ke dalam *database* tunggal.

Kemudian data hasil *extract* ini menjadi proses transformasi yang pada prinsipnya adalah mengubah kode-kode yang ada menjadi kode-kode standar, misalnya kode propinsi. Hal ini perlu dilakukan mengingat data-data yang diambil

berasal dari sumber yang berbeda yang memungkinkan memiliki standarisasi yang berbeda pula. Standarisaasi diperlukan untuk nantinya memudahkan pembuatan laporan.

Arsitektur data *warehouse* yang umum biasanya menempatkan satu *server database* terpisah yang disebut *staging* yang berfungsi untuk menangani proses *extract* dan *transform* ini sebelum dilakukan proses *load* ke tujuan akhir data *warehouse.* Proses *load* dalam ETL adalah suatu proses mengirimkan data yang telah menjalani proses transformasi ke gudang data akhir, yaitu data *warehouse* itu sendiri dimana aplikasi *reporting* dan *business intelligence* siap mengakses. Di bawah ini adalah penjelasan dari tiap-tiap proses *extract, transform,* dan *load* [4].

1. *Extract*

Bagian pertama dari suatu proses ETL aalah mengekstrak dari sumber data disebut ekstrak, akrena 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 dimsukan 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 formal 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”

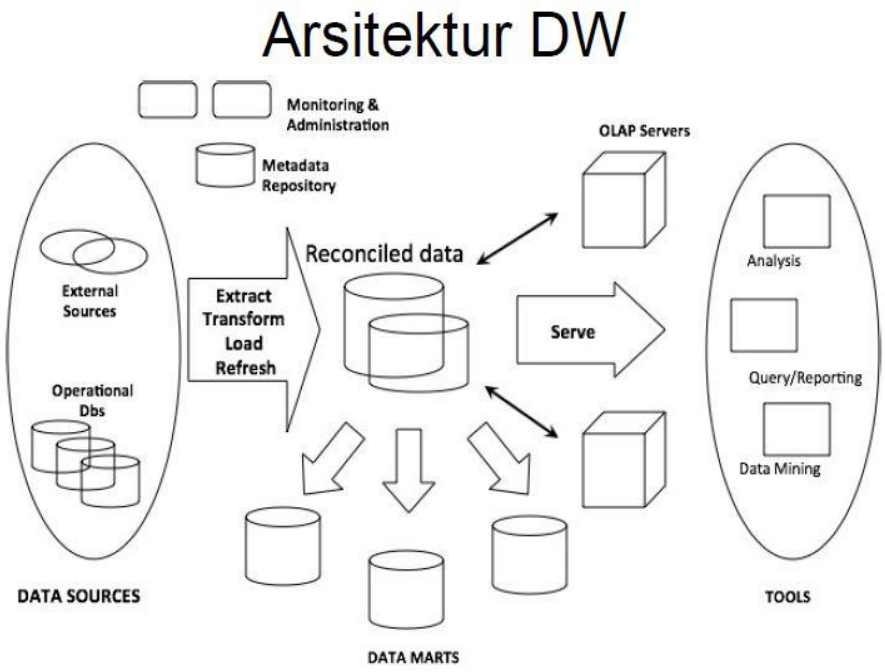
3. *Load*

Tahap *load* adalah menload data ke dalam target akhir yang apda 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.

Arsitektur data untuk data *warehouse* mempunyai kompoenen utama yaitu *read only database*. Karakteristik arsitektur data *warehouse* adalah sebagai berikut:

1. Data diambil dari sistem asal (sistem informasi yang ada), *database* dan *file*.
2. Data dari sistem asal diintegrasikan dan ditransformasi sebelum disimpan ke dalam *database management system* (DBMS) seperti oracle, mysql dan lain-lain.
3. Data *warehouse* merupakan sebuah *database* terpisah bersifat hanya dapat dibaca yang dibuat khusus untuk mendukung pengambilan keputusan.
4. Pemakai mengakses data *warehosue* melalui aplikasi *front end tool*.

Arsitektur dan komponen utama dari data *warehouse* dapat dilihat pada gambar 2.2 dibawah ini :



Gambar 2.2 Arsitektur DataWwarehouse

Gambar 2.2 menunjukkan proses data *warehouse*, dimana data *source* diolah (*extract, transform, load*) menjadi data *warehouse* untuk selanjutnya di qiuery untuk menghasilkan *report* yang diinginkan. Data *mart* berisikan kumpulan data yang digunakan untuk pengambilan keputusan. Data *mart* lebih spesifik dan ditunjukkan untuk bagian-bagian tertentu.

2.5.3 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 [5] :

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 dimesions*. Jadi dimensi menunjukkan latar belakang kontekstual dari *fact*. Bnayak proses analisisi yang digunakan untuk menghitung (*quality*) dampak dari dimensi pada *fact*.

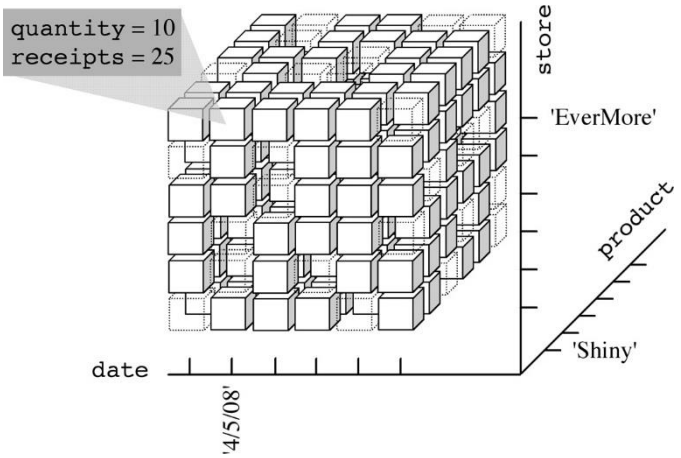
3. *Measures*

Suatu *measures* (ukuran) adalah suatu besaran (angka numerik) atribut daris ebuah *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*.

2.5.4 Visualisasi Dari Suatu Dimensional Model

Cara yang paling populer dalam menvisualisasikan suatu model dimensional aalah dengan menggambarkan sebuah *cube* (Gambar 2.3) 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, jadis ebuah kubus lebih biasa digunakan. Gambar berikut merupakan *measurent* adalah

volume dari produksi, yang mana dijelaskan dengan kombinasi dari tiga dimensi yaitu lokasi produk dan waktu. Dimensi dari lokasi dan dimensi produk mempunyai dua level hirarki. Misalnya dimensi mempunyai *level region* dan *plant*.



Gambar 2.3 Kubus suatu visualisasi dari dimensional model

Terdapat tiga konsep yang dapat memodelkan lebih jelas dalam penentuan *fact* table *dimensi dan measure*, yaitu [6] :

1. Model *star*

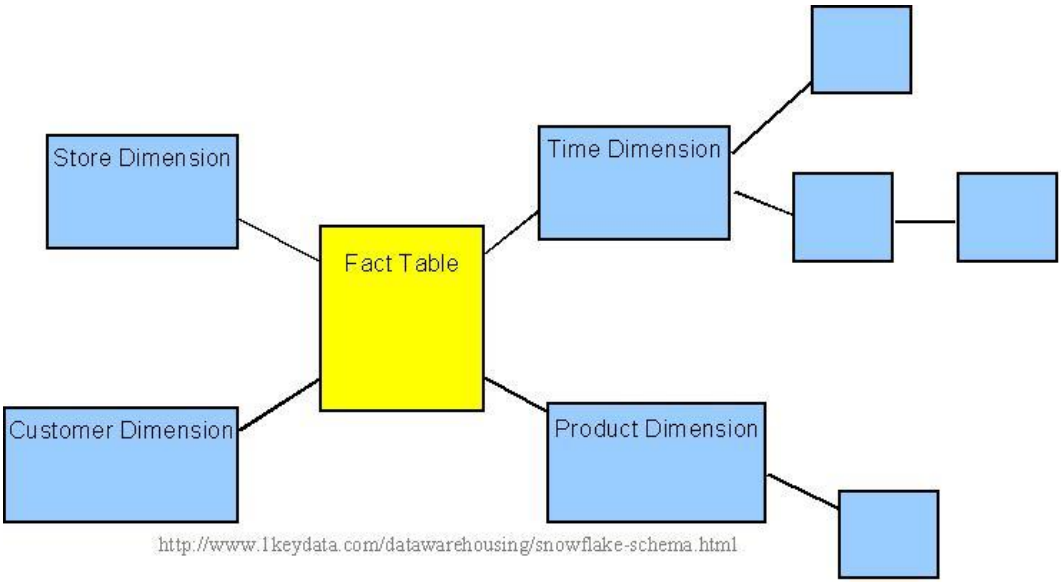
Model *star* adalah model dasar dari pemodelan multidimensi yang memiliki suatu tabel induk yang dinamakan *table fact* dan kumpulan dari tabel-tabel kecil yang disebut tabel dimensi disusun dalam pola-pola melingkar mengelilingi *table fact*. Untuk lebih jelasnya terlihat pada gambar 2.4



Gambar 2.4 Model Star

2. Model *Snowflake*

Selain *star* dalam menampilkan desain sebuah *data warehouse* juga dapat menggunakan model *snowflake*. 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. Untuk lebih jelasnya terlihat pada gambar 2.5

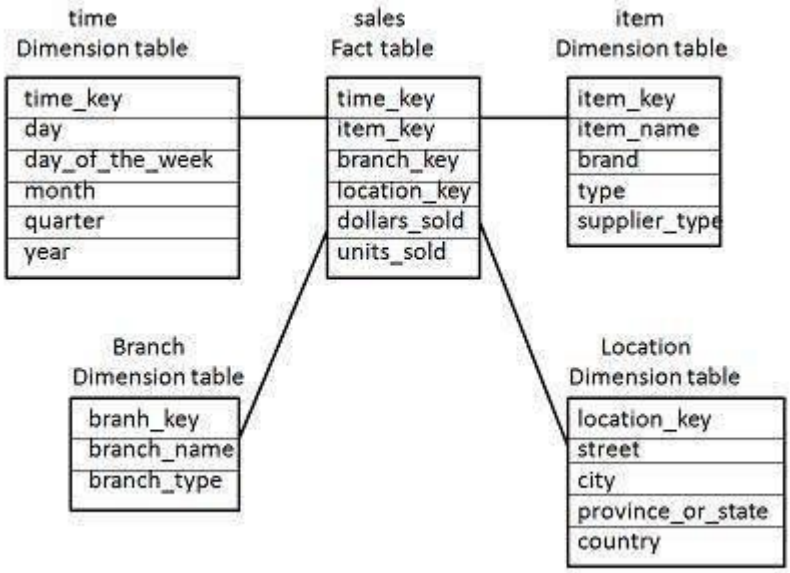


Gambar 2.5 Model Snowflake

Pada model *snowflake* diatas dapat dilihat bahwa tiap-tiap tabel dimensi berhubungan dengan dimensi yang lain seolah-olah tabel dimensi tersebut merupakan tabel fakta. Hal ini dapat terjadi karena pada model *snowflake* ini telah dilakukan normalisasi. Normalisasi ini tidak terdapat pada model *star*.

3. Model *Constellation*

Beberapa *fact table* berbagi tabel dimensi. Ditampilkan sebagai koleksi dari kumpulan skema bintang yang sering disebut sebagai skema *galaxy*. Untuk lebih jelasnya terlihat pada gambar 2.6



Gambar 2.6 Model Snowflake

2.5.5 Online Analytical Processing (OLAP)

Online analytical processing (OLAP) merupakan salah satu *tools* yag digunakan untuk mengakses informasi dalam data *warehouse*. Teknologi OLAP memungkinkan data *warehosue* 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 di evaluasi secaa cepat dengan menggunakan *online analysis* dan *grafical tool*. Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk men-support proses analisis secara *real time* [7].

2.5.5.1 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* di ekstrak secara periodik dan digunakan untuk mengupdate obyek 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 perngkat lain yang mendukung OLE DB dengan ekstensi OLAP dapat melakukan akses dan memanupulasi data yang ada.

2.5.5.2 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 [7]:

1. Multidimensional OLAP (MOALP)

Salah satu cara umum yang digunakan untuk menyimpand data adalah dalam basis data multimensional. Tidak seperti basis data relasional yang menyimpan informasi dalam sejumlah baris dalam tabel. Sebuah basis data multimensional menyimpan informasi dalam sejumlah *array* multimensional. 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 *relational* dalam data *warehosue* untuk menyimpan agregasi, sedangkan *detail* dari *fact table* tersimpan dalam data *warehouse fact table*. Banyak orang berpendapat bahwa basis data yang dirancang secara khsusus 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 rarusan ribu *recors* 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*.

2.5.5.3 OLAP *Cube* (Kubus OLAP)

Objek utama yang tersimpan dalam sebuah basis data OLAP adalah *cube* (kubus). Sebuah kubus merupakan representasi multimensi 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 [7]:

1. *Processing*

Operasi *processing* pada kubus antara lain melakukan *loading* dan *refreshing* data pada kubus yang dilakukan ole OLAP *service engine*. Tabel dimensi dibaca pertama kali untuk mengumpulkan *level* dengan anggota dari data aktual, setelah itu dilanjutkan dengan pembacaan *fact table* dan ikuti 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 ari 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)*

Consolifation atau lebih dienal 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 bnayaknya informasi yang tersedia

2.6 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,a taupun komponen aplikaisnya, dalam bentuk console, aplikasi windows,a taupun aplikasi web. Visual studion 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 megnembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas windows) ataupun *managed code* (dalam bentuk microsft intermediate language di NET framework). Selain itu, visual studion juga dapat digunakan untuk mengembangkan aplikasi silverlight, aplikasi windows mobile.

2.7 SQL Server

Microsoft SQL server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk microsoft. Bahasa kueri utamanya adalah *transast-sql* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh microsfot. Umumnya SQL server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudia 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 laind ari 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 adaah 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 minning.
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 dikirimi 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 obyek 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).