

## **BAB III**

### **LANDASAN TEORI**

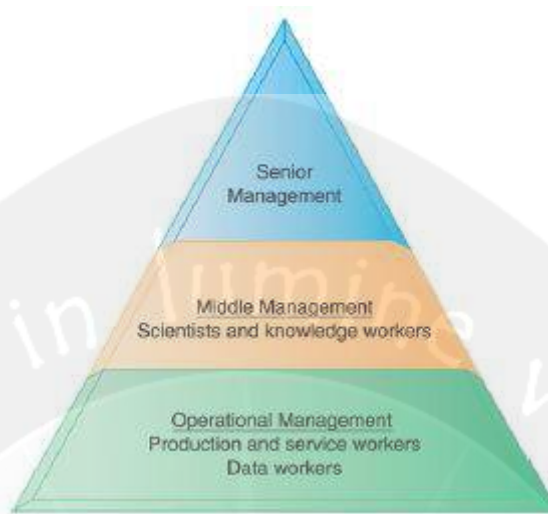
Dalam bab ini akan dijelaskan mengenai beberapa konsep tentang sistem informasi, intelegensi bisnis, *data warehouse*, *data mart*, *near real-time*, *change data capture*, keuangan dan *tools* yang digunakan.

#### **3.1. Sistem Informasi**

##### **3.1.1. Pengertian Sistem Informasi**

Sistem informasi secara teknis dapat didefinisikan sebagai seperangkat komponen yang saling terkait yang mengumpulkan (atau menerima), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi. Selain mendukung pengambilan keputusan, koordianasi, dan kendali, sistem informasi juga dapat membantu manajer dan pekerja menganalisa masalah, memvisualisasikan subjek yang kompleks, dan menciptakan produk baru. Sistem Informasi merupakan bagian integral dari organisasi.

Elemen utama dari sebuah organisasi adalah orang-orang, struktur, proses bisnis, politik, budaya. Organisasi memiliki struktur yang terdiri dari tingkat dan spesialisasi yang berbeda. Struktur organisasi mengungkapkan pembagian yang jelas mengenai tenaga kerja. Kewenangan dan tanggung jawab dalam perusahaan bisnis yang diorganisasikan sebagai hirarki atau struktur paramida. Hirarki pada tingkat atas terdiri dari manajerial, profesional dan karyawan teknis, sedangkan tingkat yang lebih rendah terdiri dari operasional personil.



Gambar 3.1. Hirarki Organisasi Bisnis (Laudon dan Laudon, 2012)

Manajemen senior membuat keputusan strategis jangka panjang tentang produk dan jasa serta memastikan kinerja keuangan perusahaan. *Middle management* melaksanakan program dan rencana manajemen senior. manajemen operasional bertanggung jawab untuk memantau kegiatan bisnis sehari-hari. *Knowledge workers*, seperti insinyur, ilmuwan, atau arsitek, desain produk atau jasa dan menciptakan pengetahuan baru bagi perusahaan, sedangkan *data workers*, seperti sekretaris atau juru tulis, membantu dengan penjadwalan dan komunikasi di semua tingkat perusahaan. Pekerja produksi atau jasa benar-benar menghasilkan produk dan memberikan layanan.

Para ahli dipekerjakan dan dilatih untuk fungsi bisnis yang berbeda. Fungsi bisnis utama, atau tugas-tugas khusus yang dilakukan oleh organisasi bisnis, terdiri dari penjualan dan pemasaran, manufaktur dan produksi, keuangan dan akuntansi, dan sumber daya manusia. Seperti yang terlihat pada tabel di bawah ini:

Tabel 3.1. Fungsi Utama Bisnis (Laudon dan Laudon, 2012)

| <b>Fungsi</b>           | <b>Tujuan</b>  |
|-------------------------|--|
| Penjualan dan Pemasaran | Menjual produk dan jasa organisasi   |
| Manufaktur dan produksi | Memproduksi dan memberikan produk dan layanan  |
| Keuangan dan Akuntansi  | Mengelola aset keuangan organisasi dan memelihara catatan keuangan organisasi                        |
| Sumber Daya Manusia     | Menarik, mengembangkan, dan mempertahankan tenaga kerja pada organisasi, memelihara catatan karyawan |

### 3.1.2. Jenis Sistem Informasi

Jenis-jenis sistem informasi menurut Laudon dan Laudon (2012), antara lain:

a. *Transaction Processing Systems* (TPS)

Sistem bisnis dasar yang mendukung tingkat operasional dalam organisasi. TPS merupakan sistem terkomputerisasi yang menjalankan dan mencatat transaksi rutin sehari-hari yang diperlukan untuk menjalankan bisnis.

b. *Management Information Systems* (MIS)

MIS mendukung fungsi-fungsi perencanaan, pengontrolan, dan pengambilan keputusan pada tingkat manajemen.

c. *Decision Support Systems* (DSS)

DSS mendukung keputusan non-rutin. DSS berfokus pada permasalahan yang unik dan cepat berubah.

d. *Executive Support Systems* (ESS)

ESS mendukung tingkat strategis dalam organisasi dengan mendukung manajer tingkat atas dalam mengambil keputusan. ESS menangani keputusan nonrutin yang memerlukan penilaian, evaluasi, dan pandangan karena tidak ada prosedur yang telah disepakati/ditentukan untuk mencapai solusi. ESS dirancang untuk menyaring,

memadatkan, dan melacak data penting, menampilkan data penting.

### **3.2. Intelegensi Bisnis**

Dalam sub bab ini akan dijelaskan mengenai pengertian intelegensi bisnis.

#### **3.2.1. Pengertian Intelegensi Bisnis**

Stackowiak et al. (2007) mendefinisikan Intelegensi Bisnis sebagai proses mengambil jumlah data yang besar, menganalisis data tersebut, dan menyajikan satu set laporan tingkat tinggi yang meringkas inti data menjadi langkah bisnis, sehingga memungkinkan pihak manajemen untuk membuat keputusan bisnis pokok sehari-hari. Cui et al. (2007) memandang BI sebagai cara dan metode untuk meningkatkan performa bisnis dengan memberikan bantuan yang kepada pengambil keputusan eksekutif untuk memungkinkan mereka mempunyai informasi yang dapat ditindaklanjuti.

Sedangkan Zeng et al (2006) mendefinisikan BI sebagai proses mengumpulkan, melaporkan, dan menyebarkan informasi yang memiliki tujuan, serta pengurangan ketidakpastian dalam pembuatan keputusan strategis. BI mencakup *data warehouse* yang efektif dan juga komponen reaktif yang mampu memantau waktu proses operasional untuk memungkinkan pengambil keputusan taksis dan operasional untuk menyesuaikan tindakan mereka sesuai dengan strategi perusahaan (Golfarelli et al, 2004).

Sementara menurut analisis dari Gartner Group, BI merupakan sebuah proses dalam mengubah data menjadi informasi. Melalui penemuan-penemuan iteratif, informasi tersebut diubah menjadi business intelligence

(intelegensi bisnis). Kuncinya adalah bahwa intelegensi bisnis merupakan sebuah proses yang *cross functional*, sesuai dengan pemikiran manajemen pada saat ini, dan tidak dihadirkan dalam istilah Teknologi Informasi (Dresner, 2002).

### **3.2.2. Data Warehouse**

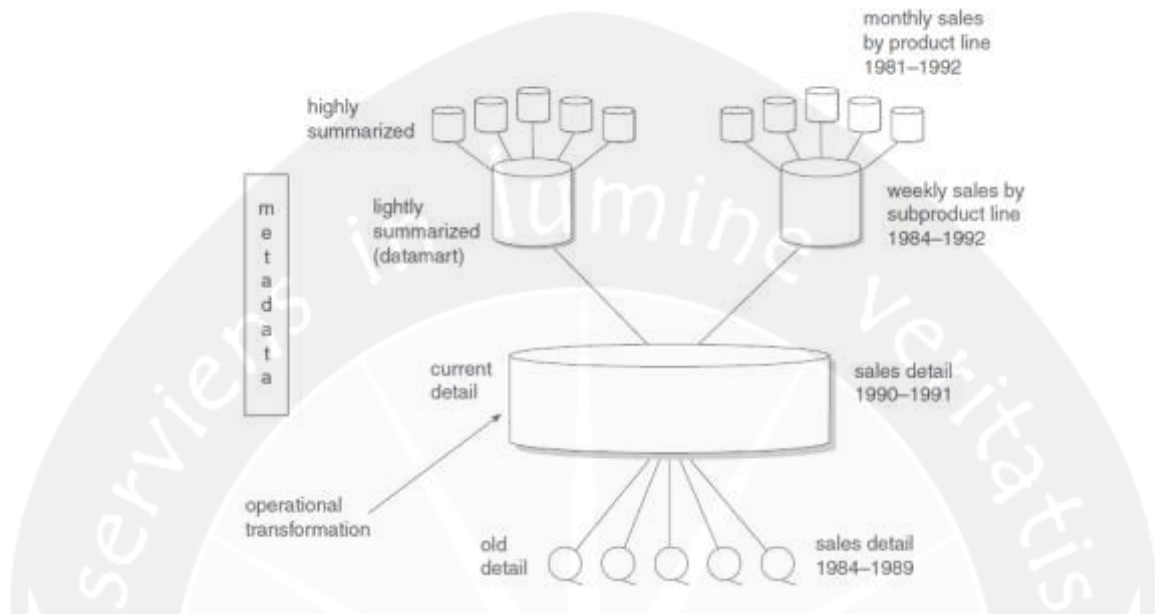
*Data warehouse* merupakan salah satu komponen dari intelegensi bisnis. Maka dalam bagian ini akan dijelaskan mengenai pengertian *data warehouse*, struktur *data warehouse*, arsitektur *data warehouse*, karakteristik *data warehouse*, dan komponen-komponen *data warehouse* dari *data warehouse* yang menjadi teori dalam penelitian ini.

#### **3.2.2.1. Pengertian Data Warehouse**

Menurut Inmon (2005), *data warehouse* adalah koleksi data yang mempunyai sifat berorientasi subjek, terintegrasi, *time variant*, dan bersifat tetap dari koleksi data dalam mendukung proses pengambilan keputusan manajemen.

### 3.2.2.2. Struktur Data Warehouse

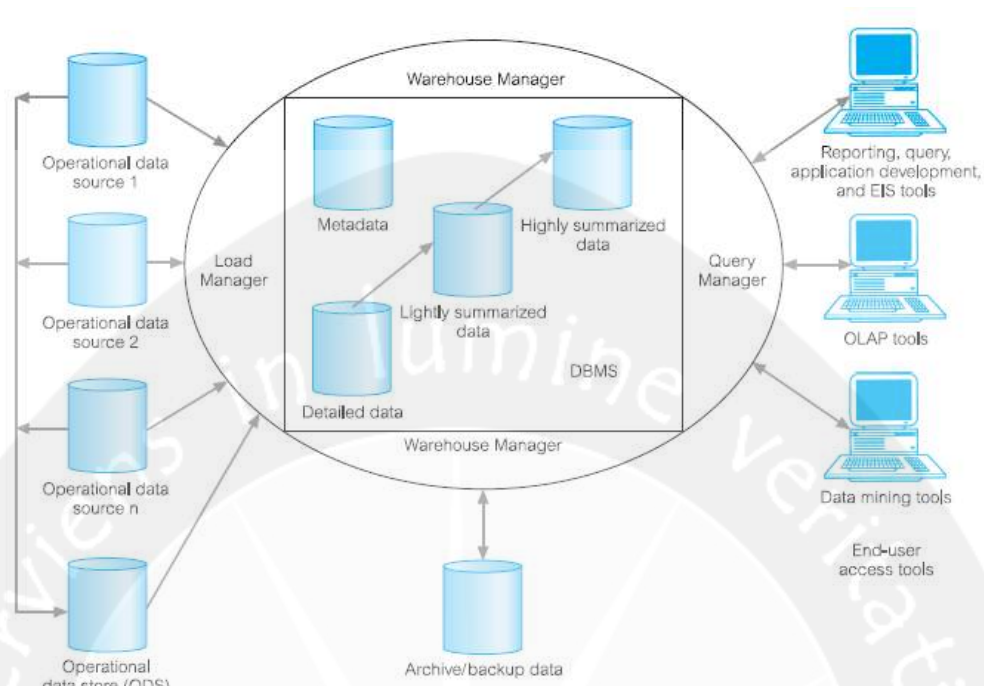
*Data warehouse* memiliki struktur sebagai berikut:



Gambar 3.2. Struktur Data Warehouse (Inmon, 2005)

### 3.2.2.3. Arsitektur Data Warehouse

Menurut Connolly dan Begg (2005), arsitektur *data warehouse* terdiri komponen-komponen yang saling menunjang satu sama lain dalam membangun *data warehouse*. Gambar di bawah ini menunjukkan arsitektur dan komponen utama dalam sebuah *data warehouse*:



Gambar 3.3. Arsitektur Data Warehouse (Connolly dan Begg, 2005)

Komponen-komponen dari data *warehouse* di atas antara lain:

1. *Operational Data*

*Operational Data* merupakan sumber data untuk *data warehouse* yang tersedia dari *mainframe* data operasional pada hirarki generasi pertama dan *network database*.

2. *Operational Data Source (ODS)*

ODS merupakan sebuah penyimpanan dari data operasional sekarang dan terintegrasi yang digunakan untuk analisis.

3. *Load Manager* (disebut juga *front end component*)

*Load Manager* akan melaksanakan semua operasi yang terkait dengan proses *extraction* dan *loading* dari data ke *data warehouse*.

#### 4. *Warehouse Manager*

*Warehouse Manager* akan melakukan seluruh operasi-operasi yang berhubungan dengan kegiatan manajemen data di dalam *warehouse*. Operasi-operasi tersebut meliputi:

- a. Analisis terhadap data untuk memastikan konsistensi.
- b. Transformasi dan penggabungan sumber data dari tempat penyimpanan sementara ke dalam tabel-tabel *data warehouse*.
- c. Penciptaan indeks-indeks dan *view* berdasarkan tabel-tabel dasar.
- d. Melakukan denormalisasi dan agregasi jika diperlukan.
- e. *Backing-Up* dan mengarsipkan data.

5. *Query Manager* (disebut juga komponen *back-end*), melakukan operasi-operasi yang berhubungan dengan manajemen *user queries*. Operasi-operasi yang dilakukan oleh komponen ini termasuk mengarahkan *query* kepada tabel-tabel yang tepat dan menjadwalkan eksekusi dari *query* tersebut.

6. *Detailed Data*, merupakan area dalam *data warehouse* yang menyimpan semua data detail di dalam *schema database*.

7. *Lightly dan Highly Summarized Data*, merupakan area dalam *data warehouse* yang menyimpan semua agregat data yang digeneralisasikan oleh *warehouse manager*.

8. *Archive / Backup Data*, merupakan area dalam *data warehouse* yang menyimpan detail dan ringkasan data dengan tujuan untuk arsip dan *backup* data.



9. *Metadata*, merupakan area dalam *data warehouse* yang menyimpan semua *metadata* (data yang menerangkan data lain) yang digunakan oleh semua proses dalam *data warehouse*. *Metadata* digunakan dalam berbagai macam tujuan, meliputi:

a. Proses *extraction* dan *loading*

*Metadata* digunakan untuk mapping sumber data ke dalam *view* yang sesuai dari data dalam *data warehouse*.

b. Proses manajemen *data warehouse*

*Metadata* digunakan untuk mengotomatiskan produksi dari tabel ringkasan.

c. Bagian dari proses manajemen *query*

*Metadata* digunakan untuk menghubungkan *query* secara langsung dengan sumber data yang dibutuhkan.

10. *End-User Access Tools*, fungsi utama dari *data warehousing* adalah untuk menyediakan informasi kepada pengguna untuk pengambilan keputusan strategis. Pengguna berinteraksi dengan *data warehouse* menggunakan *end-user access tools*. *Data warehouse* harus mendukung efisiensi *ad hoc* dan analisis rutin. Kinerja tinggi dicapai dengan perencanaan untuk keperluan *join*, *summations*, dan laporan secara berkala oleh pengguna akhir. *End user access tools* dapat dikategorikan ke dalam lima kelompok utama, yaitu :

a. *Reporting dan query tools*

b. *Application development tools*

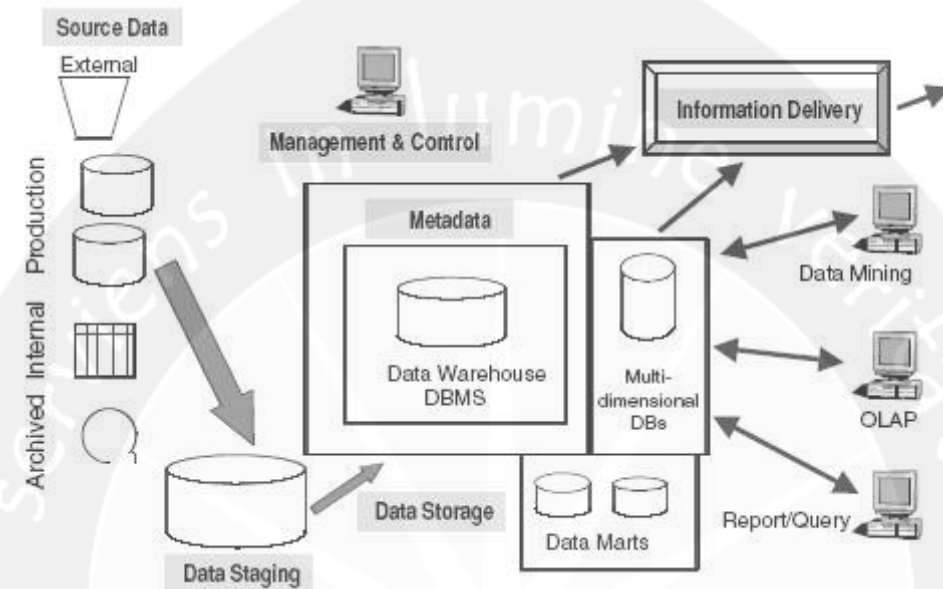
c. *Executive Information Systems (EIS) tools*

d. *Online Analytical Processing (OLAP) tools*

e. *Data mining tools*

### 3.1.2.4. Komponen Data Warehouse

Pada subbab ini, akan dibahas tentang komponen-komponen *data warehouse* menurut Ponniah (2010) yang dapat dilihat pada gambar 3.4.



Gambar 3.4. Komponen *Data Warehouse* (Ponniah, 2010)

Komponen sumber data berada di sebelah kiri. Komponen *data staging* sebagai blok pembangun berikutnya. Di tengah, dapat dilihat komponen penyimpanan data yang mengelola *data warehouse*. Komponen *information delivery* berada di sebelah kanan, yang terdiri dari semua hal menyediakan informasi dari *data warehouse* bagi pengguna.

#### 1. Komponen Sumber Data

Sumber data yang digunakan dalam *data warehouse* dapat dikelompokkan menjadi empat, yaitu:

##### a. *Production Data*

Data dalam kategori ini berasal dari berbagai sistem operasional dalam perusahaan. Berdasarkan kebutuhan informasi di dalam *data warehouse*,

segmen data dipilih dari sistem operasional yang berbeda. Dalam proses ini, data-data yang ditangani kemungkinan besar berada dalam format yang bermacam-macam, kemungkinan juga berasal dari *platform* yang berbeda-beda. Lebih lanjut lagi, data-data tersebut didukung oleh sistem basis data dan sistem operasi yang berbeda-beda. Karakteristik *production data* yang paling signifikan dan mengganggu adalah perbedaan (*disparity*).

b. *Internal Data*

Dalam setiap perusahaan, masing-masing pengguna menjaga dokumen-dokumen, *spreadsheets*, profil-profil pelanggan, terkadang bahkan basis data departemental. Inilah yang disebut dengan *internal data* atau data internal, bagian yang dapat menjadi berguna dalam sebuah *data warehouse*.

Data-data internal dalam perusahaan seperti ini tidak dapat diabaikan. Data ini menambah kompleksitas dalam proses transformasi dan integrasi data sebelum dapat disimpan ke dalam *data warehouse*. Harus ditentukan strategi-strategi untuk mengambil data dari *spreadsheets*, menemukan cara untuk mengambil data dari dokumen-dokumen tekstual, dan menghubungkannya ke dalam basis data-departemental untuk mengumpulkan data yang saling berhubungan dari sumber-sumber tersebut.

c. *Archived Data*

Sistem operasional utamanya ditujukan untuk menjalankan suatu bisnis. Di setiap sistem operasional, kita secara berkala mengambil data lama dan menyimpannya di dalam *file-file* arsip. Seperti telah disebutkan sebelumnya, sebuah *data warehouse* menyimpan

*snapshot-snapshot* data historikal. Untuk mendapatkan informasi historikal, perlu melihat ke dalam data set-data set yang telah diarsipkan. Data tipe ini berguna untuk memahami pola-pola dan menganalisis tren-tren.

#### d. *External Data*

Banyak para eksekutif menggunakan data yang berasal dari sumber-sumber eksternal untuk meningkatkan persentase informasi yang mereka gunakan. Mereka menggunakan statistik-statistik yang berhubungan dengan industry mereka, yang dihasilkan oleh agensi-agensi eksternal. Mereka menggunakan data-data *market share* dari para kompetitor, serta nilai-nilai standar dari indikator-indikator keuangan terhadap bisnis mereka untuk mengetahui kinerja perusahaan. Data eksternal ini sangat dibutuhkan apabila suatu perusahaan ingin membandingkan perusahaannya dengan organisasi lain.

## **2. *Komponen Data Staging***

Setelah mengekstraksi data dari berbagai macam sistem operasional, data harus disiapkan untuk disimpan ke dalam *data warehouse*. Data-data hasil ekstraksi yang berasal dari beberapa sumber berlainan harus diubah, dikonversi, dan membuatnya siap dalam satu format yang sesuai untuk disimpan dan digunakan bagi keperluan *query* dan analisis. Tahap pembersihan ini dikenal juga dengan istilah *Extraction, Transformation, and Loading* (ETL).

Tahap pembersihan ini berlangsung di sebuah *staging area*. *Data staging* menyediakan sebuah tempat dengan satu set fungsi untuk membersihkan, mengubah, menggabungkan, mengkonversi, mencegah duplikasi data,

dan menyiapkan data sumber untuk penyimpanan dan penggunaan dalam *data warehouse*.

#### **a. Data *Extraction***

Data sumber mungkin berasal dari mesin-mesin yang berbeda dan dalam format-format data yang berlainan. Sebagian dari sumber data mungkin berada dalam sistem-sistem basis data relasional. Beberapa data mungkin berada dalam *legacy network* dan model-model data hirarkial. Banyak data-data mungkin masih berada dalam bentuk *flat files*. Bisa juga data yang diinginkan berasal dari *spreadsheets* dan data set-data set departemental lokal. Jadi, ekstraksi data bisa menjadi benar-benar kompleks.

*Software-software* untuk ekstraksi data sudah tersedia di pasaran. Untuk beberapa sumber data dapat digunakan *software* dari luar yang sesuai. Untuk sumber-sumber data lainnya, bisa jadi digunakan program-program yang dikembangkan sendiri. Tim pengembang *data warehouse* mengekstrak sumber ke dalam lingkungan fisik yang terpisah, yang membuat pemindahan data tersebut ke dalam *data warehouse* menjadi lebih mudah. Pada lingkungan yang terpisah, sumber data dapat diekstraksikan ke dalam sebuah grup *flat file*, atau sebuah basis data relasional *data-staging*, atau kombinasi dari keduanya.

#### **b. Data *Transformation***

Setelah melalui tahap ekstraksi, data tersebut masih merupakan data mentah dan tidak dapat diaplikasikan ke *data warehouse*. Karena data-data operasional didapatkan dari banyak sistem lama, kualitas data pada sistem-sistem tersebut menjadi kurang baik untuk *data*

*warehouse*. Kualitas data harus diperkaya dan dikembangkan sebelum dapat digunakan dalam *data warehouse*. Sebelum memindahkan data yang telah diekstrak dari sumber data ke dalam *data warehouse*, diperlukan beberapa macam transformasi data. Data harus ditransformasikan menurut standar-standar yang telah ditetapkan. Secara sederhana, ini mencakup antara lain pengisian nilai-nilai yang hilang dari atribut-atribut dalam data yang telah diekstrak.

Fungsi-fungsi transformasi data dapat dibedakan menjadi beberapa tugas dasar, diantaranya:

#### 1. *Selection*

Terletak di awal proses transformasi data. Yaitu memilih seluruh atau bagian-bagian dari beberapa *record* dari sumber data.

#### 2. *Splitting/joining*

Tugas ini meliputi manipulasi data yang diperlukan untuk bagian-bagian *record* hasil operasi *selection*. Kadang-kadang data akan dipisahkan (*split*) bahkan lebih jauh lagi selama transformasi data. Operasi *join* yang dilakukan terhadap bagian-bagian dari *record* hasil operasi *selection* lebih banyak terjadi di dalam *data warehouse*.

#### 3. *Conversion*

Tahap ini meliputi konversi dari sebuah *field*, untuk dua alasan utama. Pertama untuk standarisasi data-data hasil ekstraksi dari sumber data yang berbeda-beda, dan yang kedua untuk membuat *field-field* berarti dan dapat dimengerti oleh pengguna.

#### 4. *Summarization*

Terkadang tidak memungkinkan untuk menyimpan data pada detail level terendah dalam *data warehouse*. Mungkin pengguna tidak memerlukan data pada granularitas terendah untuk analisis atau *query*, karena itu diperlukan ringkasan (*summary*) untuk disimpan ke dalam *data warehouse*.

Tipe-tipe yang transformasi data yang paling umum antara lain:

##### a. *Format Revisions*

Revisi-revisi ini meliputi perubahan tipe data dan panjang data dari *field-field* individual.

##### b. *Decoding of Fields*

Pada sumber data yang berbeda mungkin terdapat *field-field* yang mengandung kode-kode yang berbeda pula untuk merepresentasikan nilai dari *field* tersebut. Contohnya sebuah sumber data yang menggunakan angka 1 dan 2 untuk menandakan laki-laki dan perempuan.

##### c. *Calculated and Derived Values*

Tipe ini berguna untuk menghitung dan mendapatkan nilai-nilai yang diperlukan dalam *data warehouse*. Seperti misalnya pendapatan harian rata-rata atau rasio operasi.

##### d. *Splitting of Single Fields*

Contoh untuk transformasi data tipe ini misalnya memisahkan komponen nama dan alamat ke dalam *field-field* yang berbeda di dalam *data warehouse*.

##### e. *Merging of Informations*

Tipe ini tidak merupakan kebalikan dari *Splitting of Single Fields*. Data transformation tipe ini tidak secara harafiah berarti menggabungkan beberapa *field*

untuk membuat sebuah *field* data. Sebagai contoh, informasi tentang sebuah produk bisa berasal dari sumber-sumber data yang berbeda. Kode produk dan deskripsinya mungkin berasal dari sumber data yang sama, sedangkan tipe-tipe paket yang relevan dapat ditemukan dalam sumber data yang berbeda.

f. *Character Set Conversions*

Data transformasi tipe ini berhubungan dengan konversi dari set karakter menjadi set karakter standar yang telah disetujui untuk data tekstual dalam *data warehouse*. Misalnya, mengkonversi data dari sumber data dengan karakter-karakter EBCDIC ke dalam format ASCII.

g. *Conversions of Units Measurements*

Data transformasi tipe ini meliputi pengubahan ukuran sesuai dengan standar ukuran yang telah ditetapkan. Sebagai contoh jika sebuah perusahaan mempunyai cabang-cabang di beberapa negara yang menggunakan ukuran panjang berbeda (meter, kaki) maka dalam *data warehouse* harus ditentukan satu standar ukuran panjang untuk semua data.

h. *Date/Time Conversions*

Tipe ini berhubungan dengan penyajian tanggal dan waktu dalam format-format standar. Sebagai contoh adalah format tanggal Amerika dan Inggris yang akan distandarkan.

i. *Key Restructing*

*Primary keys* dari sumber data yang diekstrak akan menjadi dasar dari kunci-kunci tabel dimensi dan fakta dalam *data warehouse*.



#### j. *Deduplication*

Contoh untuk tipe ini adalah data pada *file* pelanggan. Banyak *file-file* pelanggan yang mempunyai beberapa *record* untuk pelanggan yang sama. Seringkali duplikasi disebabkan karena penambahan *record-record* akibat kesalahan.

#### c. **Data Loading**

Setelah data ditransformasikan, langkah berikutnya adalah *data loading*. Sebagian besar *data loading* mencakup pengambilan data yang telah siap (bersih), mengaplikasikannya ke *data warehouse*, dan menyimpannya ke dalam basis data yang ada disana.

Keseluruhan proses memindahkan data ke dalam *repository data warehouse* dapat dilakukan dalam beberapa cara:

##### 1. *Initial Load*

Mengumpulkan tabel-tabel *data warehouse* untuk yang pertama kalinya.

##### 2. *Incremental Load*

Melakukan perubahan-perubahan secara terus-menerus seperlunya dalam kurun waktu tertentu (secara periodik).

##### 3. *Full Refresh*

Menghapus seluruhnya isi tabel dan melakukan *reload* dengan data-data baru (*Initial Load* adalah merefresh seluruh tabel-tabel).

#### **3. Komponen Data Storage**

Penyimpanan data untuk intelegensi bisnis diletakkan pada tempat penyimpanan (*repository*) yang berbeda. Tempat penyimpanan tersebut berupa *data warehouse*, *data mart*, maupun multidimensional data. Diperlukan tempat

penyimpanan yang terpisah dari data sistem operasional. Pada umumnya suatu sistem operasional dalam perusahaan mempunyai tempat penyimpanan untuk sistem operasional yang hanya mengandung data terkini saja. Penyimpanan data untuk suatu *data warehouse* digunakan untuk menyimpan data historikal yang bervolume besar yang untuk melakukan suatu analisis.

#### **4. Komponen *Information Delivery***

Pengguna yang memerlukan informasi dari *data warehouse* antara lain:

- a. Pengguna baru: menggunakan *data warehouse* tanpa pelatihan dan karena itu memerlukan *template-template* laporan dan *query-query* yang telah ditetapkan sebelumnya.
- b. Pengguna tidak tetap: memerlukan informasi hanya sesekali, tidak secara teratur. Pengguna tipe ini juga memerlukan paket informasi yang dipersiapkan.
- c. Pengguna analis bisnis: memerlukan fasilitas untuk melakukan analisis kompleks dengan informasi dalam *data warehouse*.
- d. Pengguna yang berkemampuan: ingin dapat melihat-lihat/mengetahui seluruh *data warehouse*, mengambil data-data yang menarik, membentuk *query query* sendiri, melakukan *drill* pada lapisan-lapisan data, dan membuat laporan- laporan tertentu dan *query-query ad hoc*.

Dalam *data warehouse* mungkin diperlukan untuk memasukkan lebih dari satu mekanisme *information delivery*. Yang paling umum yaitu penyediaan *query-query* dan laporan-laporan secara *online*. Para pengguna akan menyampaikan permintaan mereka secara *online* dan menerima hasil-hasilnya secara *online* pula. Dapat juga

ditetapkan penyampaian laporan-laporan terjadwal melalui e-mail, atau dapat digunakan intranet milik organisasi yang memadai untuk penyampaian informasi. Bahkan sekarang ini penyampaian informasi melalui internet pun sudah mulai meluas.

### **3.1.3. Data Mart**

Selain *data warehouse*, dikenal juga istilah *data mart*. *Data mart* merupakan bentuk kecil atau subset dari *data warehouse* yang menyediakan informasi-informasi spesifik untuk kebutuhan-kebutuhan departemental bagi pengguna. Jadi *data mart* dapat berupa *data warehouse* yang berorientasi subjek untuk keperluan-keperluan fungsional atau departemental, atau dapat juga berupa *enterprisewide data warehouse* dalam bentuk kecil yang mengkombinasikan data dari beberapa area subjek dan bertindak sebagai sebuah titik sumber dari *enterprise data warehouse* (Adhitama, 2010).

Dalam tugas akhir ini, akan dibangun sebuah *data mart* untuk bagian keuangan UAJY. Hal ini dikarenakan bagian keuangan merupakan salah satu subjek yang ada di UAJY. Sehingga yang akan dibangun bukanlah *enterprisewide data warehouse* melainkan sebuah *data mart departmental*.

#### **3.1.3.1. Pengertian Data Mart**

Menurut Connolly dan Begg (2005), *data mart* adalah bagian dari *data warehouse* yang mendukung kebutuhan dari suatu departemen atau fungsi bisnis tertentu.

### **3.1.3.2. Karakteristik Data Mart**

Menurut Connolly dan Begg (2005), karakteristik yang membedakan data *mart* dengan data *warehouse* antara lain:

- a. Sebuah data *mart* hanya berfokus kepada kebutuhan dari pengguna yang berhubungan pada suatu departemen atau fungsi bisnis.
- b. Data *mart* berisi data yang lebih sedikit dibandingkan dengan data *warehouse*, sehingga lebih mudah dimengerti.
- c. Data *mart* biasanya tidak berisi fungsi operasional yang mendetail seperti data *warehouse*.

Ada beberapa pendekatan untuk membangun data *mart*. Salah satunya adalah dengan membangun data *mart* yang akan berkelanjutan untuk diintegrasikan menjadi sebuah data *warehouse*. Pendekatan lainnya adalah dengan membangun infrastruktur data *warehouse* perusahaan pada saat bersamaan dibangun pula satu data *mart* atau lebih untuk memenuhi kebutuhan bisnis yang berlangsung.

### **3.1.3.3. Perbedaan Data Warehouse dan Data Mart**

Perbedaan-perbedaan antara data *mart* dan data *warehouse* dapat dilihat pada tabel 3.2.

Tabel 3.2. Perbedaan *Data Warehouse* dan *Data Mart*  
(Green, 2003)

| Properti           | <i>Data Warehouse</i>               | <i>Data Mart</i>                              |
|--------------------|-------------------------------------|---|
| Lingkup            | Perusahaan<br>( <i>Enterprise</i> ) | Departemen                                    |
| Subjek             | Lebih dari satu                     | Subjek tunggal, <i>Line of Business (LOB)</i> |
| Sumber Data        | Banyak                              | Sedikit                                       |
| Ukuran             | GB sampai TB                        | Ratusan MB sampai GB                          |
| Waktu Implementasi | Bulanan sampai tahunan              | Bulanan                                       |

### 1. Lingkup

Sebuah *data warehouse* berhubungan dengan lebih dari satu area subjek dan biasanya diimplementasikan dan diatur oleh sebuah unit organisasional pusat seperti departemen IT perusahaan. Seringkali disebut dengan *data warehouse* pusat atau perusahaan. Sedangkan *data mart* biasanya hanya dibuat untuk departemen atau bagian dari perusahaan yang tertentu saja dan tidak mewakili seluruh informasi perusahaan seperti *data warehouse*.

### 2. Subjek

Sebuah *data mart* merupakan bentuk departemental dari *data warehouse* yang dirancang untuk sebuah garis bisnis tunggal (*single line of business/LOB*).

### 3. Sumber data

Sebuah *data warehouse* umumnya mengambil data dari banyak sistem sumber, sedangkan *data mart* mengambil data dari sumber-sumber yang jumlahnya lebih sedikit.

#### **4. Ukuran**

*Data mart* tidak dibedakan dari *data warehouse* berdasarkan ukuran, tetapi dalam penggunaan dan manajemen. Satu definisi dari *warehouse* yang sangat besar adalah: "Suatu *warehouse* yaitu lebih besar daripada *backup time window*."

#### **5. Waktu implementasi**

*Data mart* biasanya lebih sederhana daripada *data warehouse* dan karena itu lebih mudah untuk dibuat dan dipelihara. Sebuah *data mart* juga dapat dibuat sebagai langkah "pembuktian konsep" terhadap pembangunan sebuah *enterprisewide data warehouse*.

#### **3.1.4. Near Real-time Intelegensi Bisnis**

Pada sub bab ini akan dijelaskan mengenai *near real-time* intelegensi bisnis yang akan digunakan oleh penulis dalam penelitian ini.

BI berkembang menjadi *near real-time BI*. Dalam *near real-time BI* data akan dianalisis segera setelah memasuki organisasi. Dalam konteks ini, *near real-time* berarti memberikan informasi berkisar dari milidetik sampai beberapa detik. Latensi juga dikurangi menjadi mendekati nol.

Waktu operasional *near real-time BI* didasarkan pada *data warehouse* dengan peningkatan siklus refresh untuk memperbarui data lebih sering. *Near Real-time* dapat meng-*update* data, dimana latensi data biasanya berada dalam kisaran menit. *Near real-time BI* mengoptimalkan proses pengambilan keputusan dengan mengurangi sampai menghilangkan latensi.

Berdasarkan penjelasan ini, *near real-time* BI menyediakan fungsi yang sama seperti BI tradisional, tetapi beroperasi pada data yang diambil dari sumber data operasional dengan latensi mendekati nol dan menyediakan sarana untuk menyebarkan aksi ke proses bisnis secara *near real-time*. Secara khusus, *near real-time* BI bisa terdiri dari: pengiriman informasi secara *real-time*, pemodelan data *real-time*, analisis data *real-time*, tindakan *real-time* berdasarkan wawasan (Azvine et al., 2006).

Definisi lainnya mengenai *near real time* atau *real time data warehousing* adalah sebuah cara dari sebuah *data warehouse* untuk memperbaharui data, dari penyimpanan data statis dalam rangka memenuhi pendukung keputusan yang dibutuhkan. *Real-time* dalam sebuah BI merupakan proses dari penyaluran informasi mengenai informasi operasional dari sebuah bisnis yang terjadi dengan latensi minimal. Semua *real-time* atau *near real-time* BI memiliki latensi minimal, tetapi tujuannya adalah untuk meminimalisir waktu dari *business event* yang terjadi untuk melakukan pembenahan dan penginisialisasian.

Ketika perencanaan pada sebuah waktu yang 'tepat' maka penting halnya untuk menyesuaikan antara kebutuhan teknologi untuk waktu tindakan yang sebenarnya dibutuhkan oleh bisnis. Beberapa situasi membutuhkan aksi yang *real-time*. Dalam konteks ini *real-time* berarti jarak dari milidetik untuk beberapa detik setelah *event* sebuah bisnis terjadi (Popeanga & Lungu, 2012).

### 3.1.5. Change Data Capture (CDC)

Pada sub bab ini akan dijelaskan mengenai *Change Data Capture* yang akan digunakan oleh penulis dalam pembuatan *Near Real-Time Business Intelligence* untuk subjek keuangan UAJY. *Change Data Capture* (CDC) merupakan solusi yang sering dilakukan pada *data warehouse*. CDC adalah pendekatan inovasi untuk integrasi data, berdasarkan identifikasi, menangkap, dan mengirimkan perubahan yang dibuat oleh sumber data. Dengan memproses hanya perubahan data saja, CDC membuat proses integrasi data lebih efisien dan mengurangi biaya dengan mengurangi latensi (Attunity, 2006).

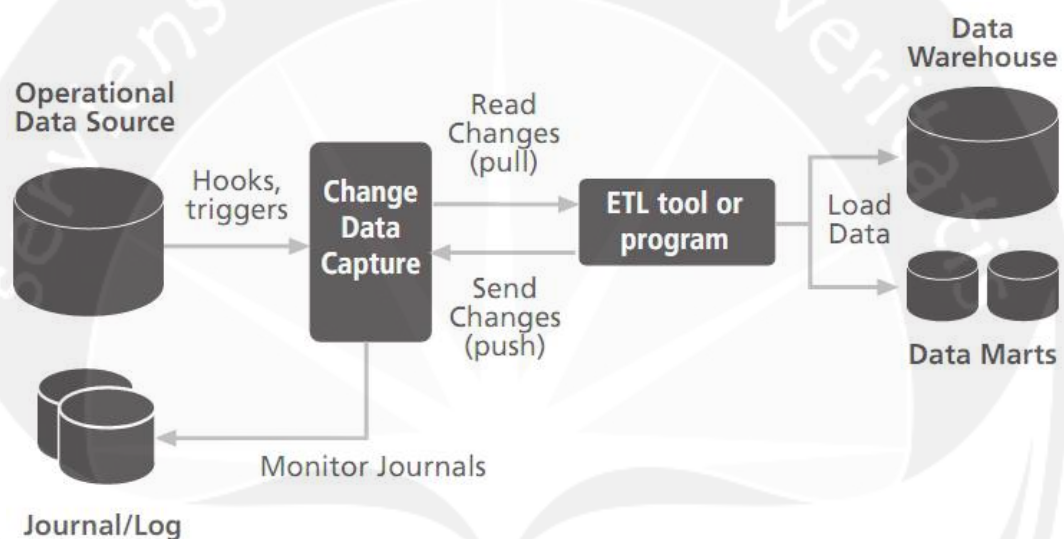
#### 1. Skenario CDC

CDC harus diintegrasikan dengan *tool* ETL sehingga proses ETL dapat efisien. Integrasi CDC dengan *tool* ETL yang ada menyediakan pendekatan terintegrasi untuk mengurangi jumlah informasi yang dikirimkan sambil meminimalisasi kebutuhan sumber daya dan memaksimalkan kecepatan dan efisiensi (Tank et.al., 2010). Terdapat dua model skenario CDC yang terintegrasi dengan *tool* ETL (Attachmate Corp, 2005):

a. Model CDC *Pull*: *Tool* ETL secara periodik meminta perubahan data, setiap waktu menerima sekumpulan *record* yang merepresentasikan semua perubahan yang ditangkap sejak siklus permintaan terakhir. Permintaan perubahan data dapat dalam frekuensi tinggi atau rendah. Skenario ini mirip dengan ETL tradisional, perbedaannya skenario ini menangkap dan memindahkan hanya data yang berubah saja. Organisasi sebaiknya menggunakan metode ini ketika kebutuhan latensinya tidak memerlukan periode waktu per menit atau per jam.



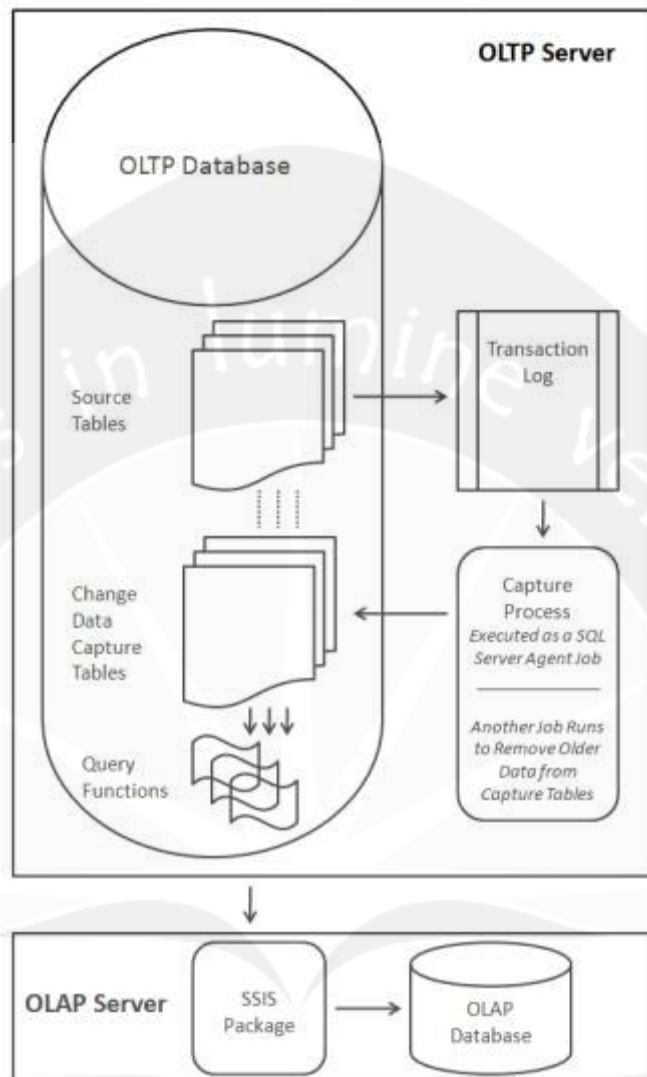
b. Model CDC *Push* : Mekanisme pengiriman mengirim perubahan data ke *tool* ETL setelah perubahan terjadi. Dapat dilakukan dengan mekanisme *eventdelivery* atau *messaging middleware*. Metode ini membutuhkan *tool* ETL menggunakan *listeners* yang menunggu *event* perubahan dan *publisher* yang digunakan untuk mengirim dan memberikan notifikasi perubahan secara *realtime*. Diagram berikut menunjukkan CDC pada proses ETL secara umum:



Gambar 3.5. Integrasi CDC pada ETL (Attachmate Corp, 2005)

## 2. CDC Pada SQL Server 2008

Pada SQL Server 2008, CDC menangkap dan merekam aktivitas *insert*, *update*, dan *delete* pada *database* OLTP dan menyimpannya dalam bentuk yang mudah digunakan oleh aplikasi, seperti *SSIS package*. *Package* ini digunakan untuk mengambil data dan menyimpannya pada server OLAP. Gambar berikut merepresentasikan gambaran komponen utama pada arsitektur CDC pada SQL Server 2008. Diagram ini dibagi menjadi dua bagian, bagian atas merepresentasikan server OLTP dan bagian bawah merepresentasikan server OLAP.



Gambar 3.6. Arsitektur CDC di SQL Server 2008 (McGehee, 2008)

Komponen-komponen CDC pada gambar di atas dijelaskan sebagai berikut (McGehee, 2008):

- Tabel sumber: ketika SQL Server pertama kali diinstal, secara *default* CDC dimatikan sehingga langkah pertama adalah mengaktifkan CDC pada level *database*, kemudian CDC harus diaktifkan pada level tabel. Setiap tabel yang CDC-nya aktif disebut dengan tabel sumber.
- Tabel CDC: Setiap tabel sumber yang CDC-nya aktif, diciptakan tabel CDC yang berhubungan, yang digunakan untuk menyimpan perubahan yang dibuat di tabel sumber,

bersama dengan beberapa metadata yang digunakan untuk menelusuri perubahan.

c. Fungsi *Query* CDC: setiap tabel sumber yang CDC-nya aktif, beberapa fungsi *query* CDC diciptakan untuk mengakses tabel CDC.

d. *Capture* dan *Cleanup Jobs*: dua SQL Server Agent *jobs* juga diciptakan, yaitu *Capture* dan *Cleanup Job*. *Capture job* secara umum berjalan terus menerus dan digunakan untuk memindahkan perubahan data ke tabel CDC dari *transaction log*. *Cleanup job* dijalankan terjadwal untuk menghapus data lama pada tabel CDC sehingga tidak membengkak terlalu besar.

Cara kerja CDC sesuai dengan Gambar 3.18, adalah sebagai berikut (McGehee, 2008): ketika ada statemen *insert*, *update*, atau *delete* yang terjadi pada tabel sumber, perubahan ini ditulis pada *transaction log* pada *database*. Hal ini normal, terjadi pula walaupun CDC tidak diaktifkan. Perbedaannya, ketika CDC diaktifkan pada tabel sumber, SQL Server Agent *Capture job* membaca perubahan pada *transaction log* dan memindahkannya ke tabel CDC yang tepat.

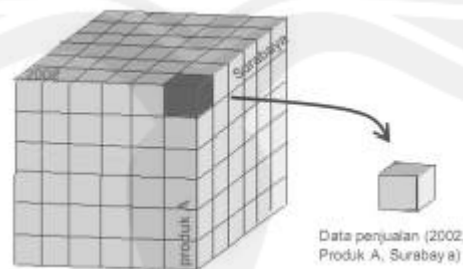
*Insert* dan *delete* masing-masing menghasilkan satu baris pada tabel CDC, dan *update* menghasilkan dua baris: satu baris data sebelum dan satu baris data setelah perubahan. Perubahan ini terus diakumulasikan pada tabel CDC sampai dihapus oleh *Cleanup Job*.

Perubahan data dari tabel CDC diekstrak dengan menjalankan statemen yang sesuai menggunakan fungsi *query* yang relevan. Sebagai contoh, SSIS *package* dieksekusi setiap empat jam. Statemen yang sesuai dengan fungsi *query* akan dijalankan, yang mengijinkan

untuk mengekstrak semua perubahan yang dibuat pada tabel sumber sejak terakhir kali SSIS *package* dieksekusi, dan kemudian memindahkannya ke *database* pada server OLAP. Dengan cara ini, data pada server OLAP selalu sinkron dengan data pada server OLTP, dengan hanya empat jam penundaan.

### 3.1.6. **Pemodelan Multidimensional**

Pembuatan *data warehouse* didasarkan pada model data multidimensi. Model ini menampilkan data dalam bentuk kubus. Data multidimensi adalah ketika sebuah data dapat dipandang dari berbagai sudut. Pusat dari objek metadata pada multidimensional adalah *cube* atau kubus yang mengandung hubungan struktur dimensi, hirarki, level dan anggota. Misalnya pada hasil penjualan suatu barang dipandang dari dimensi waktu, lokasi, pembeli dan lain-lain. Sehingga jika digambarkan, sumbu x mewakili dimensi waktu, sumbu y mewakili dimensi produk dan sumbu z mewakili dimensi lokasi. (Prasetyo et.al., 2010).



Gambar 3.7. Data Multidimensi (Prasetyo et.al., 2010)

Komponen model multidimensional yang umum ditemukan dalam perancangan *data warehouse* (Prasetyo et.al., 2010):

#### a. Dimensi

Dimensi merupakan sebuah kategori yang independen dari multidimensional basis data. Tipe dari dimensi ini mengandung item yang digunakan sebagai kriteria *query* untuk ukuran basis data. Contoh pendistribusian busana di suatu daerah: dimensi Daerah = {Jawa Timur, Jawa Barat, Jawa Tengah, Sumatra Selatan, Surabaya, Bandung, Jakarta, Palembang, Dago, Caringin, Senen, Matraman}, dimensi Waktu = { tahun 2003, tahun 2004, tahun 2005, bulan Januari, bulan Februari, bulan Maret, hingga bulan Desember, tanggal 1, tanggal 2, tanggal 3, tanggal 4, hingga tanggal 28/29/30/31}, dimensi Busana = { Koko, Daster, Kaos, celana panjang, celana pendek, kerudung, dll}.

#### b. Tabel Fakta

Tabel fakta merupakan pusat dari *schema* pada OLAP. Tabel fakta mempunyai dua tipe kolom, yaitu kolom yang menyimpan nilai-nilai numerik atau yang biasa disebut dengan *measure* dan kolom yang *menyimpan foreign key* yang mengacu ke tabel dimensi. Nilai numerik yang ada pada tabel fakta merupakan nilai agregat dari data yang berasal dari tabel dimensi. Hubungan antara tabel fakta dengan tabel dimensi adalah *one to many*, sehingga masing-masing *primary key* dari tabel dimensi dijadikan *key* acuan pada tabel fakta. Dengan demikian, tabel fakta menyimpan setiap kombinasi *key* tabel dimensi yang melingkupinya.

#### c. Measure

Nilai *measure* terletak pada tabel fakta. *Measure* juga cerminan dari fakta dan juga mengandung data yang akan dianalisa seperti contoh pada gambar 3.8. OLAP

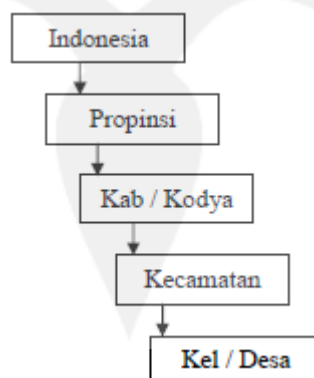
memerlukan informasi kolom bertipe numerik yang akan dijadikan *measure*.

| Sales Fact Table |           |
|------------------|-----------|
| time_key         |           |
| item_key         |           |
| location_key     |           |
| unit_sold        | } measure |
| avg_sales        |           |
| dollars_sold     |           |

Gambar 3.8. *Measure* dari Tabel Fakta

#### d. Hirarki

Hirarki merupakan bentuk kesatuan dari dimensi. Sebuah dimensi bisa terbentuk dari multilevel, yang mempunyai *parent-child relationship*. Hirarki didefinisikan bagaimana hubungan antar level. Sebagai contoh pada dimensi Daerah, hirarki mungkin akan melakukan agregasi data dari tiap level-level. Seperti pada gambar 3.9.



Gambar 3.9. Hirarki Dimensi Daerah (Adhitama, 2010)

#### e. Level

Level merepresentasikan sebuah posisi pada hirarki. Level mengumpulkan data untuk agregasi dan digunakan untuk proses komputasi. Sebagai contoh pada dimensi Daerah pada contoh sebelumnya, level yang mungkin didefinisikan adalah level Kel./Desa, level kecamatan, level Kab./Kodya, level propinsi, dan level Negara. Setiap level di atas level terendah merupakan agregasi dari level dibawahnya. Jika data dari pelanggan disimpan dalam format Kelurahan atau Desa (level Kel./Desa), maka data dapat di agregasi sesuai dengan level yang ada diatasnya, seperti level Kecamatan, level Kab/Kodya, level Propinsi dan level Negara.

#### f. Attribute

*Attribute* merepresentasikan informasi tambahan pada sebuah level tertentu. Sebuah level dapat memiliki lebih dari satu *attribute*, tetapi minimal harus memiliki satu *attribute*. Nilai *attribute* berguna sebagai nilai yang akan mewakili level ketika data multidimensi ditampilkan kepada pengguna. Hal ini Indonesia Propinsi Kab / Kodya Kecamatan Kel / Desa disebabkan tidak semua nilai pada level bisa dimengerti dan dipahami oleh pengguna. Misalnya, level *product name* menyimpan nilai *product id*, sedangkan *attribute*-nya menyimpan nilai dari *product brand*, dengan demikian yang akan ditampilkan kepada pengguna ketika pengguna memilih level *product name* adalah nilai pada kolom *product brand*, bukan *product id*.

#### g. Cube

*Cube* adalah obyek OLAP yang tersusun dari *measure*, dimensi dan *attribute*. Sisi-sisi pada *cube* ditentukan

oleh masing-masing dimensi yang terlibat dalam *cube* itu. *Cube* memiliki sisi-sisi yang menggambarkan dimensi-dimensi yang terlibat didalamnya, yang paling banyak ditemui adalah dalam bentuk tiga dimensi yang mewakili sisi baris, sisi kolom dan sisi page. Misalnya dimensi waktu, dimensi lokasi, dan dimensi produk.

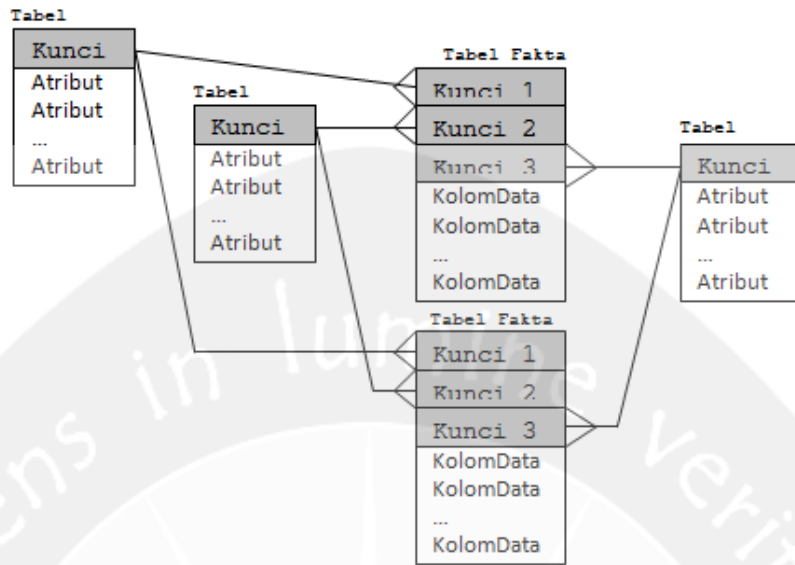
Model dimensional yang sering digunakan pada *data warehouse* adalah *star* atau *snowflake* yang mudah dimengerti dan sesuai dengan kebutuhan bisnis, mendukung *query* sederhana dan menyediakan performa *query* yang superior dengan meminimalisasi tabel-tabel join (Prasetyo et.al., 2010).

#### **3.1.6.1. Skema Bintang (*Star Schema*)**

Menurut Connolly dan Begg (2005), skema bintang adalah suatu struktur logical yang memiliki sebuah tabel fakta berisikan data-data fakta ditengahnya, dan dikelilingi oleh tabel-tabel dimensi yang berisikan data-data referensi (bisa dalam bentuk denormalisasi). Skema bintang dapat meningkatkan kecepatan, karena denormalisasi informasi ke dalam sebuah table dimensi.

Denormalisasi sangat disarankan bila terdapat sejumlah entitas yang berhubungan dengan tabel dimensi yang sering diakses, untuk menghindari perlunya menghubungkan tabel lain untuk mengakses atribut tersebut. Namun denormalisasi tidak disarankan jika data tambahan tidak sering diakses, karena dalam kasus itu menghubungkan tabel lain tidak akan terlalu memperlambat performa *query*.

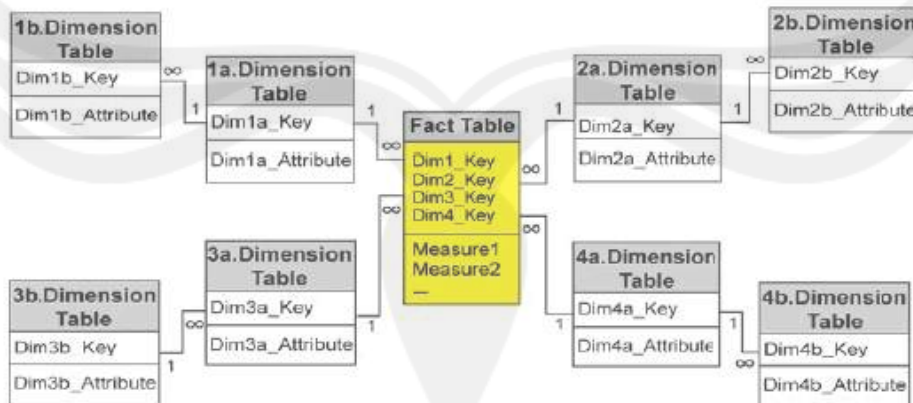




Gambar 3.10. Contoh *Star Schema* (Adhitama, 2010)

### 3.1.6.2. Skema Snowflake (Snowflake Schema)

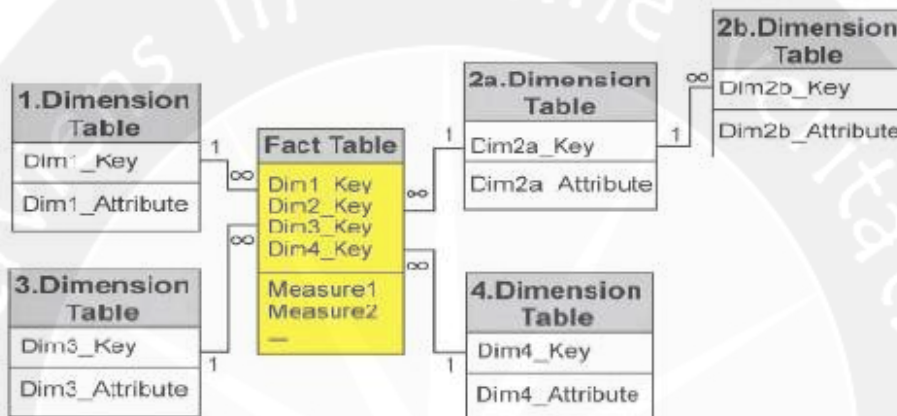
Menurut Connolly dan Begg (2005), skema *snowflake* adalah variasi dari skema bintang dimana tabel dimensi tidak berisikan data hasil denormalisasi. Sehingga tabel dimensi dapat memiliki dimensi lagi.



Gambar 3.11. Contoh *Snowflake Schema* (Connolly dan Begg 2005)

### 3.1.6.3. Skema Starflake (Starflake Schema)

Menurut Connolly dan Begg (2005), skema *starflake* adalah struktur campuran antara skema bintang dan skema *snowflake*. Beberapa tabel dimensi dapat disajikan dalam bentuk skema bintang maupun skema *snowflake* untuk memenuhi kebutuhan *query* yang berbeda-beda.



Gambar 3.12. Contoh *Starlake Schema* (Connolly dan Begg 2005)

### 3.1.7. OLTP dan OLAP

OLTP dan OLAP adalah model-model dari basis data, yang mempunyai tujuan dan kegunaan berbeda. OLTP adalah bentuk dari sistem informasi tradisional yang lebih dahulu dikenal untuk membantu operasi-operasi harian, sedangkan OLAP muncul karena dirasakannya kebutuhan akan sebuah sistem informasi yang berorientasi untuk membantu pengambilan keputusan.

#### 3.1.7.1. OLTP (Online Transactional Processing)

Berikut ini merupakan salah satu definisi OLTP: *OLTP adalah sistem yang memproses suatu transaksi secara langsung melalui komputer yang terhubung dalam jaringan (Febrian,2004).*

OLTP ditujukan untuk mendukung proses-proses transaksi harian dari sebuah organisasi. Sistem ini sangat berguna untuk membuat roda bisnis terus berputar karena digunakan untuk menangani proses-proses operasional dari perusahaan. Tipikal dari OLTP ini adalah untuk menangani sejumlah besar transaksi, yang dilakukan oleh sejumlah besar pengguna secara simultan dengan cepat (*real-time*). Beberapa aplikasi OLTP antara lain: *electronic banking, order processing, employee time clock systems, dan e-commerce*. OLTP mempunyai karakteristik yaitu transaksi hanya mengakses sebagian kecil basis data, pemutakhiran relatif sering dilakukan, serta transaksi singkat dan sederhana.

#### **3.1.7.2. OLAP (Online Analytical Processing)**

Berikut ini adalah sebuah definisi dari OLAP: *OLAP adalah sintesis dinamik, analisis dan konsolidasi dari data multidimensional yang sangat besar (Codd et al, 1993).*

OLAP diperkenalkan oleh E. F. Codd yang merupakan bapak basis data *relational*. Secara mendasar OLAP adalah suatu metode khusus untuk melakukan analisis terhadap data yang terdapat dalam media penyimpanan data (basis data) dan kemudian membuat laporannya sesuai dengan permintaan pengguna (Hermawan, 2005). OLAP adalah sebuah pendekatan secara cepat menyediakan jawaban-jawaban terhadap *query* analitik yang multidimensi di dalam basis data. OLAP merupakan bagian dari kategori yang lebih global dari pemikiran bisnis, yang juga merangkum hubungan antara pelaporan dan penggalian data. Aplikasi khusus dari OLAP adalah pelaporan bisnis untuk penjualan, pemasaran, manajemen

pelaporan, manajemen proses, penganggaran dan peramalan, laporan keuangan dan bidang-bidang yang serupa (Santosa et.al., 2011)

Beberapa operasi OLAP yaitu (Han & Kamber 2006):

- 1) *Drill up (roll-up)* ringkasan data, yaitu dengan menaikkan konsep hirarki atau mereduksi dimensi.
- 2) *Drill down (roll-down)* kebalikan dari *roll-up*, yaitu melihat data secara lebih detail atau spesifik dari level tinggi ke level rendah.
- 3) *Slice and dice*, *slice* adalah pemilihan pada satu dimensi dari kubus data yang bersangkutan dan *dice* mendefinisikan *subcube* dengan memilih dua dimensi atau lebih.
- 4) *Pivot (rotate)* memvisualisasikan operasi yang merotasikan sumbu data dalam *view* sebagai alternatif presentasi data.

### 3.1.7.3. Perbedaan Sistem OLAP dan OLTP

Tabel 3.3. menunjukkan perbedaan system OLAP dan OLTP menurut Vercellis.

Tabel 3.3. Tabel Perbedaan Sistem OLAP dan OLTP  
(Vercellis, 2009)

| <b>Karakteristik</b>      | <b>OLAP</b>                        | <b>OLTP</b>                              |
|---------------------------|------------------------------------|--|
| <b><i>Volatility</i></b>  | Dinamis                            | Statis                                   |
| <b><i>Timeliness</i></b>  | Data sekarang                      | Data sekarang dan data historis          |
| <b>Dimensi waktu</b>      | Implisit dan saat Ini              | Eksplisit dan variant                    |
| <b><i>Granularity</i></b> | Data detail                        | Data yang teragregasi dan terkonsolidasi |
| <b>Frekuensi refresh</b>  | Terus-menerus dan <i>Irregular</i> | Periodik dan regular                     |
| <b>Aktivitas</b>          | Berulang                           | Tidak dapat diprediksi                   |
| <b>Fleksibilitas</b>      | Rendah                             | Tinggi                                   |
| <b>Performa</b>           | Tinggi                             | Lambat untuk query yang kompleks         |
| <b>Tipe pengguna</b>      | Pekerja                            | <i>Knowledge workers</i>                 |
| <b>Fungsi</b>             | Operasional                        | Analitikal                               |
| <b>Tujuan Penggunaan</b>  | Transaksi                          | Query kompleks dan pendukung keputusan   |
| <b>Prioritas</b>          | Performa tinggi                    | Fleksibilitas tinggi                     |
| <b>Ukuran</b>             | Megabytes hingga Gigabytes         | Gigabytes hingga terabytes               |

### 3.2. Keuangan

Pada sub bab ini akan dijelaskan secara singkat mengenai pengertian keuangan, laporan keuangan, dan kantor keuangan yang menjadi objek penelitian dalam penyusunan tugas akhir ini.

### **3.2.1. Pengertian Keuangan**

Keuangan merupakan ilmu dan seni dalam mengelola uang yang mempengaruhi kehidupan setiap orang dan setiap organisasi. Keuangan berhubungan dengan proses, lembaga, pasar dan instrument yang terlibat dalam transfer uang diantara individu maupun antara bisnis dan pemerintah (Sundjaja & Barlian, 2002).

### **3.2.2. Laporan Keuangan**

Laporan keuangan merupakan data yang dapat memberikan informasi yang relevan bagi investor, kreditur atau pihak lain dalam mengambil keputusan ekonomi. Suatu informasi bisa dikatakan relevan apabila adanya informasi tersebut bisa membuat perbedaan keputusan yang diambil. Informasi yang relevan bisa membantu pemakai informasi untuk membentuk harapan atau kesimpulan mengenai hasil-hasil pada masa yang lalu, sekarang dan masa mendatang (Hanafi & Halim, 2005). Laporan keuangan disusun dengan tujuan untuk menyediakan informasi yang menyangkut posisi keuangan, kinerja dan perubahan posisi keuangan suatu perusahaan yang bermanfaat bagi sejumlah besar pemakai dalam pengambilan keputusan (Prastowo & Juliyanty, 2005). Laporan keuangan pada dasarnya merupakan hasil refleksi dari sekian banyak transaksi yang terjadi dalam perusahaan. Transaksi dan peristiwa yang bersifat finansial dicatat, digolongkan dan diringkas dengan setepat-tepatnya (Jumingan, 2006).

Laporan Keuangan suatu perusahaan dapat memberikan suatu informasi yang bermanfaat bagi pemakainya, jika memenuhi persyaratan yang ditetapkan (Prinsip Akuntansi Indonesia) adalah sebagai berikut:

### 1. Relevan

Pengukuran relevansi suatu informasi harus dihubungkan dengan penggunaannya. Oleh karena dalam mempertimbangkan relevansi suatu informasi hendaknya perhatian difokuskan pada kebutuhan umum pemakai dan bukan kebutuhan khusus pihak tertentu.

### 2. Dapat dimengerti

Bentuk laporan keuangan dan istilah yang dipakai hendaknya diseuaikan dengan batas pengertian pemakai informasi juga diharapkan mempunyai dasar pengertian mengenai aktivitas ekonomi perusahaan, proses akuntansi dan istilah yang digunakan dalam laporan keuangan.

### 3. Objektif

Laporan keuangan harus disusun seobyek mungkin, dapat diuji kebenarannya oleh para pengukur yang independent dan menggunakan metode pengukuran yang sama.

### 4. Netral

Laporan keuangan hendaknya disusun untuk kebutuhan umum pemakai dan bukan kebutuhan pihak tertentu saja.

### 5. Tepat Waktu

Laporan keuangan harus disampaikan secara sedini mungkin agar dapat digunakan sebagai dasar untuk membantu pengambilan keputusan ekonomi dan untuk menghindari tertunda pengambilan keputusan bagi pemakai.

### 6. Dapat Dibandingkan

Laporan keuangan yang disajikan harus dapat dibandingkan dengan laporan keuangan periode sebelumnya dari perusahaan yang sama maupun dengan perusahaan yang sejenis pada periode yang sama.

## 7. Lengkap

Laporan keuangan hendaknya disajikan secara lengkap meliputi semua data akuntansi yang memenuhi sekurang-kurangnya enam persyaratan tersebut (Sangkala, 2008).

### **3.2.3. Kantor Keuangan Universitas Atma Jaya Yogyakarta**

Visi:

Menjadi unit yang siap memberikan layanan terbaik dan professional kepada pengguna layanan dengan semangat kerendahan hati, kejujuran, keramahan, cepat dan akurat guna mendukung perkembangan, kinerja dan citra positif lembaga.

Misi:

1. Memberikan layanan pada mahasiswa, orangtua dan unit-unit pengguna di Bidang Keuangan, Akuntansi dan Pengendalian.
2. Mendukung kelancaran tugas unit-unit kerja dalam pencapaian tujuan lembaga.
3. Menopang tuntutan perkembangan lembaga, melalui pengembangan staf di bidang pengetahuan, ketrampilan dan kepribadian.

Tugas Pokok:

1. Mengelola pengembangan administrasi keuangan universitas.
2. Mengidentifikasi strategi pengembangan administrasi keuangan.
3. Mengelola pengendalian administrasi keuangan.
4. Mengelola proses pencairan anggaran.



Organisasi Kantor Keuangan terdiri dari 2 Bagian yaitu:

#### 1. Bagian Operasional

Tugas pokok dan fungsi Bagian Operasional adalah: melaksanakan pengadministrasian, dana yang berasal dari mahasiswa, melaksanakan administrasi anggaran rutin universitas, melaksanakan penerimaan, penyimpanan, pengeluaran, pembukaan dan pertanggungjawaban keuangan. Bagian Operasional terdiri atas:

- a. Sub Bagian penerimaan dan
- b. Sub Bagian pengeluaran

#### 2. Bagian Akuntansi dan Pengendalian

Bagian Akuntansi dan pengendalian mempunyai tugas pokok dan fungsi menyusun dan membuat anggaran tahunan universitas, melakukan verifikasi dan pencatatan permohonan anggaran, meneliti dan menguji kebenaran setiap bukti penerimaan dan pengeluaran, menyajikan data perkembangan pelaksanaan anggaran rutin, menyusun dan menyampaikan laporan keuangan baik bulanan, triwulan dan tahunan. Bagian Akuntansi dan Pengendalian dibantu 1 Sub, yaitu Sub. Bagian Pajak.

### **3.3. *Tools* yang digunakan**

Pada sub bab ini akan dijelaskan mengenai *tools* yang digunakan dalam penyusunan tugas akhir ini.

#### **3.3.1. SQL Server 2008**

SQL Server 2008 adalah teknologi yang mendukung *development* dan administrasi dari *Business Intelligence Application*.

#### **3.3.1.1. SQL Server 2008 Integration Services (SSIS)**

*SQL Server Integration Services* atau biasa disingkat dengan SSIS adalah sebuah *tools* yang digunakan untuk melakukan proses *Extract, Transform, dan Load* (ETL) dan diklasifikasikan sebagai fitur *Business Intelligence* (BI). Dalam kaitannya dengan BI, SSIS adalah fitur yang digunakan untuk menarik data dari ERP, *relational database*, atau *file* untuk kemudian hasilnya disimpan ke dalam data *warehouse*.

#### **3.3.1.2. SQL Server 2008 Analysis Services (SSAS)**

*Analysis Services* adalah teknologi untuk OLAP (*Online Analytical Processing*) dan *Data Mining*. Proses administrasi OLAP dilakukan di *SQL Server Management Studio* berupa *Viewing Data*, membuat *multidimensional Expression* (MDX), *Data Mining Extension* (DMX) dan *XML for Analysis* (XML/A) dan mendefinisikan *role* untuk akses *security* OLAP data (Wirama dan Sudianto dan Hermawan, 2008).

#### **3.3.1.3. Report Portal**

Report Portal adalah sebuah aplikasi klien yang menyediakan akses ke berbagai sumber data melalui internet. Aplikasi ini membutuhkan *software* lain pada sisi klien selain Internet Explorer 5.5 atau lebih tinggi, Firefox, Safari atau Chrome.

Aplikasi ini menawarkan sebuah peran berbasis model keamanan. Pengguna dan peran dapat menetapkan laporan dan folder sebagai pemberi persetujuan, pembaca atau sebagai penerbit. *Microsoft Analysis Services* mengizinkan keamanan untuk diterapkan pada level data *cell*. Untuk mengimplementasikannya, *Analysis Services* harus diatur dengan benar.

Aplikasi ini ditulis menggunakan teknologi Microsoft: JavaScript AJAX, ASP.NET, VB.NET, dan Microsoft SQL Server. Aplikasi ini juga menggunakan Microsoft XML untuk Analisis pada akses data OLAP. Report Portal menyediakan akses ke Microsoft SQL Server Analysis Services OLAP data menggunakan laporan sebagai berikut:

1. *OLAP Report*

Sebuah *pivot* tabel yang memungkinkan pengguna mendesain laporan online dengan *men-drag* dan *men-drop* *measures*, *dimensions* dan *level*.

2. *Microsoft OWC (Office Web Components) Report*

Sebuah *pivot* tabel yang memungkinkan pengguna untuk mendesain laporan online dengan *men-drag* dan *men-drop* *measure*, *dimensi*, dan *level*. Laporan ini membutuhkan komponen web Microsoft Office yang diinstal pada komputer klien. Aplikasi ini akan mendeteksi dan membantu komponen yang perlu diinstal.

3. *Analysis Services 2005 KPI Report*

Laporan ini memungkinkan pengguna untuk melihat *Key Performance Indicator* untuk dibuat dalam *Analysis Services 2005*.

4. *Data Mining Report*

Laporan ini memungkinkan pengguna untuk *membrowse* model penambangan data yang dibuat dalam *Analysis Services*.

Report Portal menyediakan Visualisasi data laporan untuk data OLAP menggunakan laporan sebagai berikut:

1. *Pie-Chart Report*

*Pie-Chart Report* memiliki kemampuan untuk memvisualisasikan berbagai dimensi pada satu halaman.

## 2. *Bar-Chart Report*

*Bar-Chart Report* memiliki kemampuan untuk memvisualisasikan berbagai dimensi pada satu halaman. Metode visualisasi informasi ini mengizinkan adanya lebih dari satu *measure* yang dapat dipilih sebagai lawan pada *measure* tunggal untuk *Pie-Chart Tree*.

## 3. *Tree-Map Report*

*Tree-Map Report* memiliki kemampuan untuk memvisualisasikan banyak data pada satu halaman. Pengguna dapat memilih dua level (detail dan grouping) dan dua *measure* (ukuran dan warna).

## 4. *Chart Grid Report*

*Chart Grid Report* memiliki kemampuan untuk melihat banyak grafik pada satu halaman. Pengguna dapat memilih tiga level (untuk data points, baris, dan kolom) dan banyak *measure*.

Report Portal juga menyediakan akses ke Sumber Data Relasional menggunakan *Report* sebagai berikut:

### 1. *ROLAP Report*

Sebuah tabel *pivot* yang mengizinkan pengguna untuk mendesain laporan online dengan *men-drag* dan *men-drop measure, dimension* dan *level*.

### 2. *SQL Report*

Sebuah laporan tabular yang dirancang secara online dengan menggunakan *SQL Builder*.

### 3. *Crystal Report*

Laporan ini mengizinkan tampilan *Crystal Report*.

### 4. *Reporting Services Report*

Laporan ini mengizinkan desain dan menampilkan laporan *Microsoft Reporting Services*.

Report Portal juga menyediakan laporan lain yang berguna untuk membantu untuk mengorganisasikan informasi:

2. *Dashboard Report*

Laporan *Dashboard* menempatkan beberapa laporan pada satu halaman web.

3. *Key Performance Indicators (KPI) Report*

Laporan KPI memberikan kemampuan melakukan desain dan melihat laporan KPI.

4. *File Upload*

Setiap jenis *file* dapat di-*upload* ke dalam aplikasi. File tersebut disimpan dalam *database*.

5. *Hyperlink*

Sebuah *hyperlink* dapat dibuat ke halaman eksternal dan memiliki semua properties laporan. *Hyperlink* bisa menjadi folder web.

6. *Data Entry Form*

7. Wiki

8. Blog