

LECTURE NOTES

7023T Advanced Database System

Session 02

Data Warehouse Components

LEARNING OUTCOMES

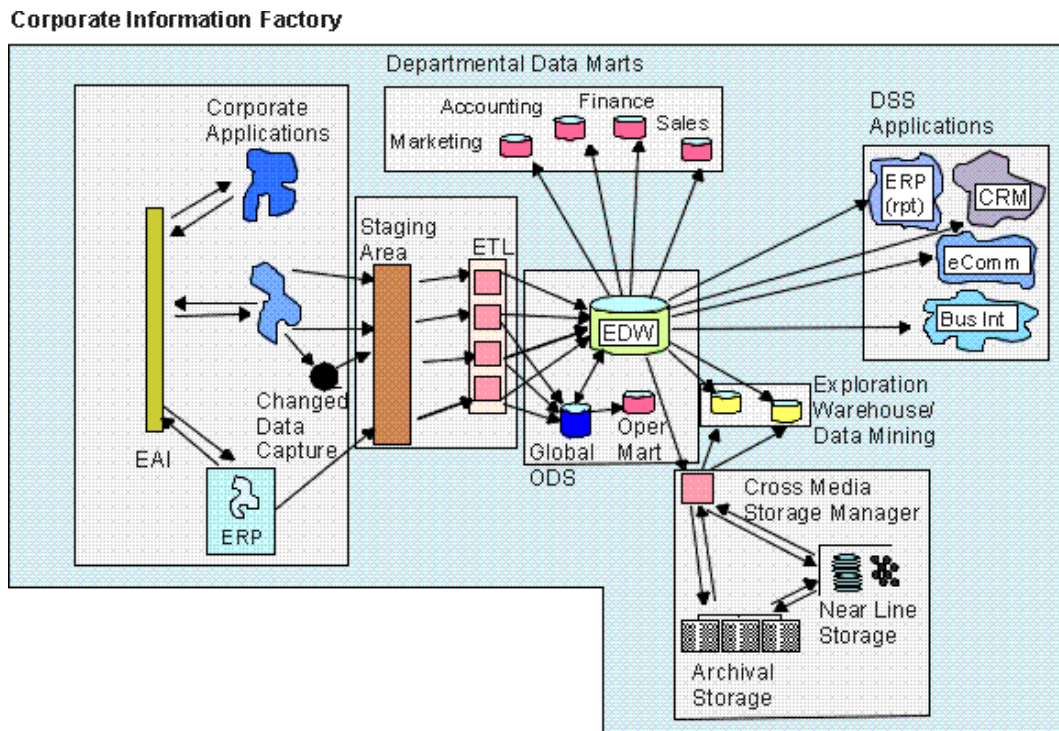
1. Peserta diharapkan memahami setiap komponen data warehouse dalam konsep *Corporate Information Factory* dan keterkaitannya satu sama lain.
2. Peserta diharapkan memahami lima jenis arsitektur data warehouse dan mengidentifikasi perbedaan utama diantara arsitektur-arsitektur tersebut.

OUTLINE MATERI (Sub-Topic):

1. *Corporate Information Factory (CIF)*
2. *External World and Applications*
3. *Integration and Transformation Layer*
4. *Extract Transform Load (ETL)*
5. *Operational Data Store (ODS)*
6. *Enterprise Data Warehouse (EDW)*
7. *Data Marts*
8. *Data Warehouse Architecture*

Corporate Information Factory (CIF)

Seperti yang sudah dibahas pada sesi kuliah sebelumnya, data warehouse muncul karena struktur dari data operasional tidak sesuai dengan kebutuhan pembuatan laporan atau analisis – khususnya untuk jangka waktu yang cukup lama. Oleh karena itu diperlukan proses transformasi terhadap data operasional agar sesuai dengan kebutuhan tersebut. *Corporate Information Factory* (CIF) yang digagas oleh Inmon, merupakan referensi arsitektur data warehouse. CIF menjelaskan sistem dan komponen yang dibutuhkan untuk mengembangkan data warehouse. Gambar 1 mengilustrasikan data warehouse sebagai kumpulan aktifitas yang mencakup mengoleksi, mengekstrak, melakukan transformasi, dan menyimpan data untuk kebutuhan pembuatan laporan maupun analisis.



by Bill Inmon and Claudia Imhoff
Copyright ©2001, all rights reserved.

Gambar 1. *Corporate Information Factory (CIF)*

External World and Applications

CIF memanfaatkan data dari dunia luar atau *external world* kemudian mengolahnya menjadi informasi yang juga berada di *external world*. Oleh karena itu karakteristik dari *external world* seperti taksonomi dari transaksi yang perlu ditangani atau jenis informasi konsumen yang perlu disediakan, merupakan informasi yang penting dalam merancang sebuah CIF. CIF mengoleksi data dari *external world* melalui komponen aplikasi. Aplikasi berinteraksi secara langsung dengan produsen data, mengumpulkan data mentah dan menyuntingnya untuk mendapatkan kualitas yang memadai. Banyak aplikasi yang tidak dirancang sebagai bagian dari suatu CIF, akibatnya aplikasi berjalan tidak optimal. Selain itu, seringkali banyak aplikasi yang tidak terintegrasi satu sama lain, masing-masing aplikasi bekerja dengan produsen data yang berbeda. Akibatnya data yang dihasilkan oleh komponen aplikasi tidak terintegrasi, kemungkinan akan terjadi duplikasi dan inkonsistensi dalam hal *key*, definisi, *encoding*, dan lain-lain.

Integration and Transformation Layer

Kita mungkin bertanya-tanya mengapa tidak menyalin data ke data warehouse daripada menggunakan aplikasi sebagai input. Sebagian mungkin berpendapat mengapa tidak mengintegrasikan aplikasi ke dalam data warehouse daripada menambahkan komponen antarmuka yang menghubungkan aplikasi dengan data warehouse. Kedua pendekatan tersebut tidak efisien apabila diterapkan pada organisasi berskala besar yang sistem informasinya dibangun pada beberapa platform berbeda. Sebagai contoh, aplikasi Human Resource dibangun di atas sistem ERP (*Enterprise Resource Planning*) seperti SAP, sedangkan untuk mendukung proses training digunakan sistem pihak ketiga, sementara untuk proses *Point of Sale* (kasir) disimpan pada database Oracle. Untuk skenario seperti ini maka dibutuhkan sebuah layer yang berfungsi untuk mengintegrasikan data dari berbagai sumber berbeda, selain itu data tersebut seringkali perlu ditransformasikan dengan alasan integritas dan konsistensi data. Disinilah layer Integration and Transformation memegang peranan penting pada *Corporate Information Factory*.

Extract Transform Load

Proses ETL terdiri dari ekstraksi (membaca data dari satu atau lebih database), transformasi (mengubah data diambil dari bentuk sebelumnya menjadi bentuk yang perlu sehingga dapat ditempatkan ke dalam sebuah data warehouse atau hanya lain data base), dan pemuatan (memasukkan data ke dalam data warehouse). Transformasi terjadi dengan menggunakan aturan atau tabel lookup atau dengan menggabungkan data dengan data lainnya. Ketiga fungsi database terintegrasi ke dalam satu alat untuk menarik data dari satu atau lebih database dan menempatkan mereka ke dalam database lain, mengkonsolidasi dengan database atau data warehouse.

Perangkat ETL juga berfungsi memindahkan data dari sumber ke target, mendokumentasikan bagaimana elemen data (misalnya, metadata) berubah, ketika mereka bergerak antara sumber dan target, pertukaran metadata yang diperlukan dengan aplikasi lain, dan mengadministrasikan semua proses *runtime* dan operasional (misalnya, penjadwalan, manajemen kesalahan, audit log, statistik), sehingga proses ETL merupakan bagian penting bagi data warehousing. Tujuan dari proses ETL adalah untuk memuat warehouse dengan data terintegrasi dan dibersihkan. Data yang digunakan dalam proses ETL dapat berasal dari berbagai sumber seperti aplikasi pada mainframe, aplikasi ERP, perangkat CRM, flat file, spread sheet Excel, atau bahkan antrian pesan.

Operational Data Store

Operational Data Store (ODS) menyediakan bentuk layanan file informasi pelanggan (*Customer Information File* - CIF). ODS merupakan jenis database yang sering digunakan sebagai pengenalan untuk data warehouse. Berbeda dengan bentuk statis dari data warehouse, isi dari ODS dapat diperbarui melalui operasi bisnis. Biasanya ODS digunakan untuk keputusan jangka pendek. melibatkan misi - aplikasi kritis bukan dan bukan untuk jangka menengah maupun keputusan jangka panjang. ODS yang mirip dengan memori jangka pendek dan menyimpan hanya informasi yang sangat baru. Sebagai perbandingan, data warehouse jangka panjang menyimpan informasi yang lebih permanen. ODS mengkonsolidasikan data dari beberapa sumber mendekati real-time, pandangan terintegrasi yang *volatile*, data yang paling mutakhir.

Enterprise Data Warehouse

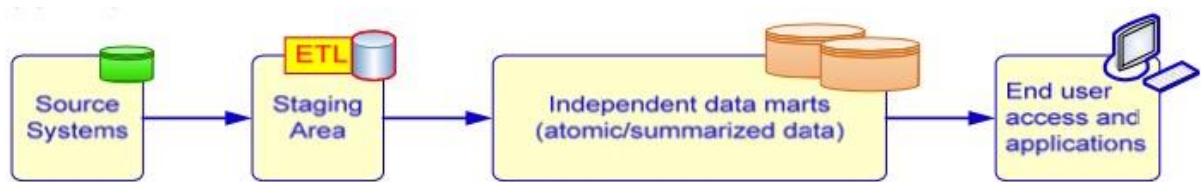
Enterprise Data Warehouse (EDW) adalah data warehouse berskala besar yang digunakan di seluruh perusahaan untuk mendukung proses pengambilan keputusan. Sifat skala besar berarti menyediakan integrasi data dari berbagai sumber ke dalam format standar untuk digunakan dalam sebuah sistem *Business Intelligence* yang efektif dan untuk aplikasi pendukung pengambilan keputusan. EDW digunakan untuk menyediakan data untuk berbagai jenis *Decision Support System* (DSS), termasuk *Customer Relationship Management* (CRM), *Supply Chain Management* (SCM), *Business Performance Management* (BPM), *Business Activity Monitoring* (BAM), *Product Lifecycle Management* (PLM), *Revenue Management* (RM), dan kadangkala bahkan sampai *Knowledge Management System* (KMS). Banyak perusahaan yang sudah mendapatkan berbagai keuntungan dari EDW, bila dirancang dan diimplementasikan secara benar.

Data Mart

Apabila data warehouse biasanya menggabungkan database di seluruh perusahaan, *data mart* mencakup lingkup yang lebih kecil dan berfokus pada subjek atau departemen tertentu. Sebuah *data mart* bisa merupakan subset dari data warehouse yang terdiri dari area subjek tunggal (misal pemasaran, operasional dan lain sebagainya). Suatu *data mart* dapat merupakan sistem yang mandiri atau dependen. Dependen *data mart* adalah subset yang dibuat langsung dari data warehouse. Hal ini memberikan keuntungan dari menggunakan model data yang konsisten dan menyediakan data yang lebih berkualitas. Suatu *data mart* mendukung konsep model data tunggal namun data warehouse tetap harus dibangun terlebih dahulu. Dependen *data mart* memastikan bahwa pengguna akhir melihat versi yang sama dari data yang diakses oleh semua pengguna data warehouse lainnya. Tingginya biaya pembuatan data warehouse membatasi penggunaannya untuk perusahaan-perusahaan besar saja. Namun sebagai alternatif, banyak perusahaan menggunakan biaya yang lebih rendah untuk berinvestasi pada versi data warehouse dengan skala kecil yang disebut sebagai *data mart* independen. Sebuah data mart independen adalah data warehouse kecil yang dirancang untuk unit bisnis strategis atau departemen, dan sumbernya bukanlah suatu EDW.

Data Warehouse Architecture

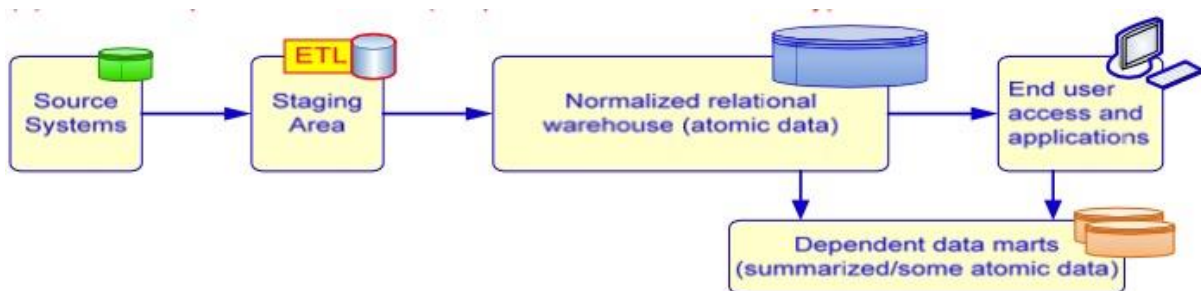
Sudut pandang disain arsitektur data warehouse secara umum dapat dikategorikan ke dalam disain data warehouse *Enterprise-wide* atau *Datamart System* (Golfarelli dan Rizzi, 2009). Aryachandra dan Watson (2006b) menjelaskan lima arsitektur data warehouse seperti diilustrasikan pada gambar 2 sampai dengan gambar 6. Perbedaan utama dari kelima arsitektur tersebut adalah pada *data marts*. Pada arsitektur *Independent Data Marts* masing-masing *data marts* berdiri sendiri dan tidak ada kaitan diantaranya. Pada arsitektur *Data Mart Bus Architecture with Linked Dimensional Data marts*, *data marts* dihubungkan oleh dimensi yang digunakan bersama atau disebut sebagai *conformed dimension*. Pada arsitektur *Hub and Spoke*, setiap data mart memanfaatkan data dari *corporate data warehouse*. Pada arsitektur *Centralized Data Warehouse*, tidak terdapat *data marts* sehingga pengguna langsung mengakses *corporate data warehouse*. Pada arsitektur *Federated*, terdapat komponen pemetaan data yang memanfaatkan input dari data warehouse/*data marts* yang sudah ada.



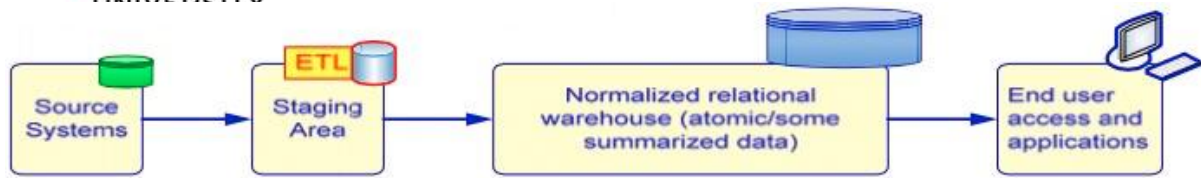
Gambar 2. *Independent Data Marts Architecture*



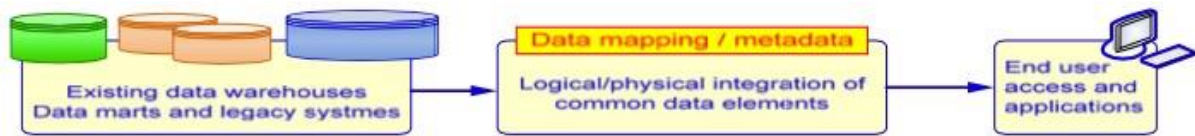
Gambar 3. *Data Mart Bus Architecture with Linked Dimensional Data marts*



Gambar 4. *Hub and Spoke Architecture*



Gambar 5. *Centralized Data Warehouse Architecture*



Gambar 6. *Federated Architecture*

SIMPULAN

- Sebuah sistem data warehouse terdiri beberapa komponen yang saling terkait. Meskipun tidak semua data warehouse harus memiliki semua komponen tersebut, namun konsep Corporate Information Factory yang diusulkan oleh Inmon merupakan model ideal dari sebuah data warehouse.
- Terdapat lima jenis arsitektur data warehouse, perbedaan utama terletak pada data marts-nya.

DAFTAR PUSTAKA

1. Kimball, R. (2008). *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons.
2. Kimball, R., & Ross, M. (2011). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons.
3. Inmon, W. H. (2005). *Building the Data Warehouse*. John Wiley & Sons.