

# Mengenal Data Warehouse

**Djoni Darmawikarta**

djoni\_darmawikarta@yahoo.ca

## ***Lisensi Dokumen:***

*Copyright © 2003 IlmuKomputer.Com*

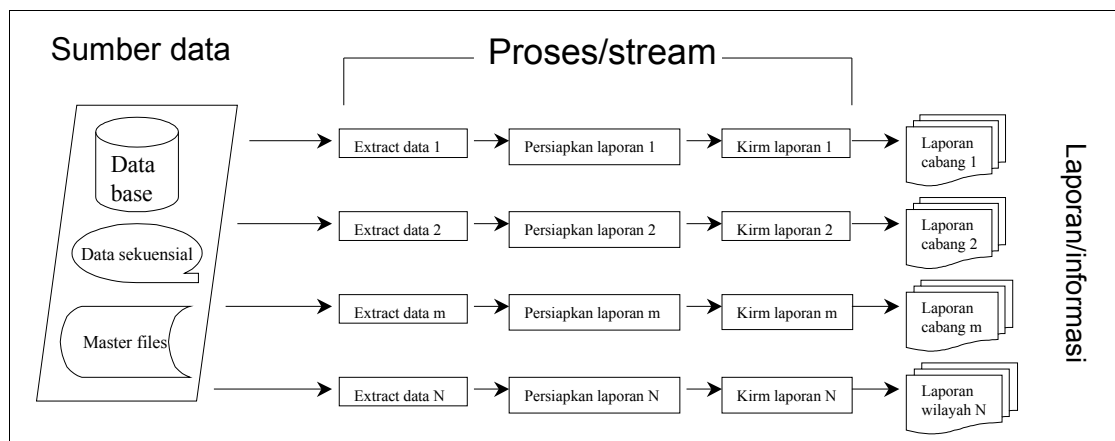
*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

## **Memanfaatkan hubungan reflexive**

Saya masih ingat, betapa suksesnya kami (waktu itu saya masih di Indonesia) mengerjakan proyek yang sama untuk sejumlah bank.; yaitu memproses penutupan akhir hari dipusat komputer (sistem adalah terpusat, centralized), menyampaikan informasi (laporan) kesemua cabang untuk buka kantor disetiap esok hari kerja. Laporan berisi misalnya posisi rekening nasabah yang hari sebelumnya mengadakan transaksi. Juga disiap- dan disebar-kan laporan sejenis untuk kantor wilayah yang mencakup cabang-cabang dibawah naungannya, dan seterusnya berantai mengikuti jenjang otonomi sistem perkantornya.

Laporan ini berjenis-jenis, meskipun semuanya menggunakan sumber data yang sama. Selain format (layout), strukturnya juga bervariasi. Ada yang diurutkan berdasar besarnya transaksi, jenis laporan lain berdasar tipe rekening, dan untuk kantor wilayah misalnya diurutkan dan dikelompokkan per kantor cabang.

Selain penggunaan sumber data yang sama, ada sebagian proses yang sama diulang untuk setiap cabang (job stream sama sekali terpisah), terutama proses untuk mengambil data yang dibutuhkan (extraction) dari database milik sistem operasional bank (yang disiang harinya dipakai untuk menangani segala macam transaksi, baik atas permintaan pelanggan melalui kantor cabang, ATM maupun internet banking, juga proses administrasi oleh bank sendiri). Situasi ini dapat digambarkan sebagai berikut.

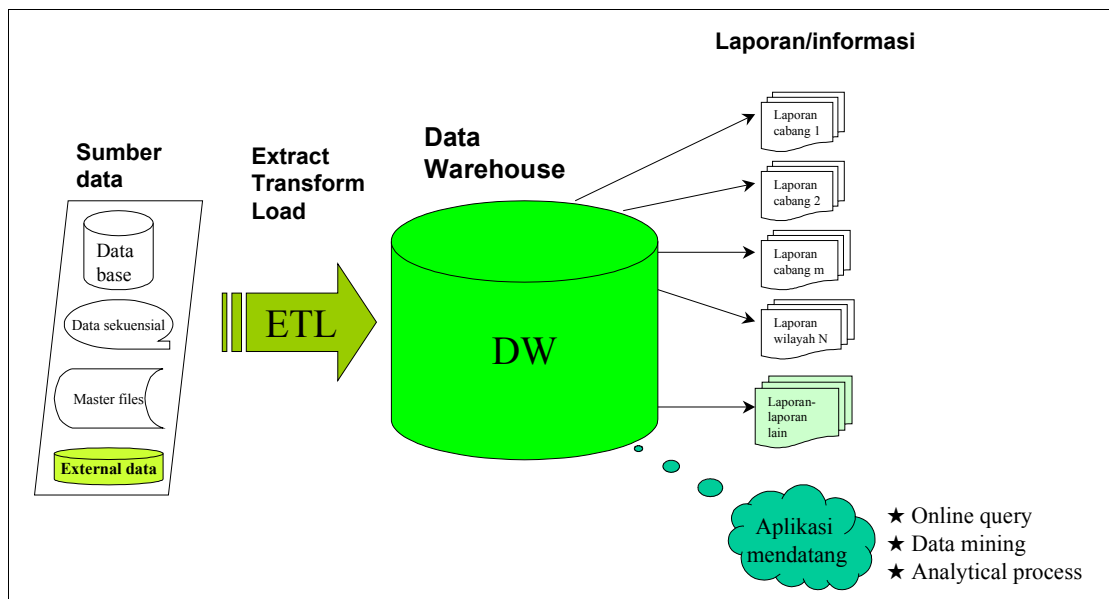


Gambar 1. Satu sumber data dengan masing-masing proses/stream untuk setiap keperluan

Mengapa mengambil dan memproses data yang sama berulang kali? Pertanyaan inilah yang menjadi alasan utama mengapa data warehouse diperlukan.

Ada alasan lain, akibat situasi diatas, yang selalu menimbulkan keraguan pemakai tingkat-atas (senior manager, misalnya kepala wilayah) akan kebenaran informasi yang diperolehnya. Laporan harian pada contoh bank-bank diatas dijadikan sumber data oleh fungsi-fungsi dikantor cabang dan wilayah, untuk membuat laporan yang lain untuk keperluan masing-masing. Akibat berantainya penggunaan dan pemrosesan individu ini (terutama re-entry manual, perbedaan pengertian nilai data (definisi dan makna) dan formula perhitungan), maka bila mereka mengadakan rekonsiliasi kembali berdasar laporan individu tersebut (misalnya waktu rapat manajemen antar fungsi), hampir selalu muncul masalah inconsistency informasi.

Sistem gudang data (Data Warehouse) dibangun untuk mengatasi masalah teknis dan bisnis dalam kasus-kasus sejenis diatas, yaitu kasus kasus yang berkaitan dengan penggunaan data dan informasi untuk mengambil keputusan bisnis dan manajemen. Bandingkan gambar 2 berikut ini dengan gambar 1.



Gambar 2.

Data hanya diambil dan diproses sekali, disimpan didalam data warehouse (proses ini disebut ETL = Extract, Transform, Load); hanya bagian proses yang unik dari setiap pemakaian data yang dilaksanakan masing-masing (biasanya menggunakan software khusus – specialized tool)

Fungsi utama Data Warehouse adalah: Mengambil (termasuk data dari luar yang dibutuhkan, misalnya daftar kode pos dari kantor pos), mengumpulkan, mempersiapkan (transforming, seperti membersihkan, mengintegrasikan, decoding), menyimpan (loading), dan menyediakan data untuk pemakai atau aplikasi yang bersifat query/reporting (read-only); hanya satu data terpercaya ini yang digunakan oleh semua yang membutuhkan (single version of truth), untuk pelaporan, analisa informasi dan mengambil keputusan (analytical application)

Seperti digambarkan di Gambar 2, sekali data masuk kedalam data warehouse, data yang memang dirancang dan ditujukan untuk bukan hanya satu atau sejumlah pemakaian yang sudah diketahui, dapat digunakan untuk aplikasi mendatang dan belum pernah sebelumnya terpikirkan- bandingkan ini dengan pembangunan aplikasi operasional (fungsional)!

### Empat karakteristik data warehouse

Ada 4 sifat yang mencirikan data yang disimpan didalam data warehouse – ini didefinisikan oleh salah satu mahaguru data warehousing (Bill Inmon)

**Subject oriented:** Aplikasi untuk operasi perusahaan (operational system) berorientasi pada proses (mengotomasi fungsi-fungsi dari proses bersangkutan – function oriented). Misalnya di bank, aplikasi kredit mengotomasi fungsi-fungsi: verifikasi lamaran dan credit checking, pemeriksaan kolateral, approval, pendanaan, tagihan, dan seterusnya. Didalam data warehouse data-data yang dihasilkan dari proses kredit ini, diatur kembali (dikelompokkan) dan diintegrasikan (digabung) dengan data-data dari fungsi-fungsi lain, agar berorientasi pada misalnya nasabah dan produk.

**Integrated:** Data dari macam-macam aplikasi transaksi (untuk bank misalnya: tabungan, kredit, rekening koran) semua mengandung data nasabah, ada yang sama ada yang spesifik (yang sama misalnya: nama dan alamat, yang spesifik misalnya: untuk kredit ada kolateral, untuk rekening koran ada overdraft) – didalam data warehouse data-data yang sama harus diintegrasikan disatu database, termasuk misalnya diseragamkan formatnya (sederhana tetapi paling sering terjadi – aplikasi-aplikasi sering dibeli vendor berbeda, dibuat dengan/dijalankan di teknologi berbeda-beda)

Time variant: Data warehouse menyimpan sejarah (historical data). Bandingkan dengan kebutuhan sistem operasional yang hampir semuanya adalah data mutakhir! Waktu merupakan tipe atau bagian data yang sangat penting didalam data warehouse. Didalam data warehouse sering disimpan macam-macam waktu, seperti waktu suatu transaksi terjadi/dirubah/dibatalkan, kapan efektifnya, kapan masuk ke komputer, kapan masuk ke data warehouse; juga hampir selalu disimpan versi, misalnya terjadi perubahan definisi kode pos, maka yang lama dan yang baru ada semua didalam data warehouse kita. Sekali lagi, data warehouse yang bagus adalah yang menyimpan sejarah!

Non-volatile: Sekali masuk kedalam data warehouse, data-data, terutama data tipe transaksi, tidak akan pernah di update atau dihapus (delete)

Terlihat, bahwa keempat karakteristik ini saling terkait – kesemuanya harus diimplementasikan agar suatu data warehouse bisa efektif memiliki data untuk mendukung pengambilan-keputusan. Dan, implementasi keempat karakteristik ini membutuhkan struktur data dari data warehouse yang berbeda dengan database sistem operasional.