

# **Bab 1: Data Warehousing**

## **1.1 Pendahuluan**

Saat sekarang kita lihat bahwa URLs merupakan bagian dari kehidupan kita sehari-hari dan tukar-menukar alamat email sudah merupakan hal yang biasa. Bisnis saat ini memungkinkan kita dapat membeli atau menjual segala sesuatu lewat web. Kita dapat menemukan toko terdekat, atau restoran paling bagus. Kita selalu mendapatkan informasi melalui pesawat telpon wireless atau menaruh catatan di PDAs. Kita tidak akan pernah tersesat dengan adanya GPS yang bisa memandu. Kita gunakan aplikasi self-service baik di tempat kerja maupun di rumah. Kita bisa mentransfer uang dari satu rekening ke rekening yang lain dengan sekali klik dan tidak perlu lagi khawatir untuk mengakses bank pada saat jam kerja yang sibuk atau tidak amannya mesin ATM di malam hari.

Sebenarnya ada apa di belakang semua teknologi ini ? Hampir selalu diperlukan database untuk mengorganisasi dan menyimpan data, dan seringkali diperlukan data warehouse untuk mendukung proses pengambilan keputusan dalam bisnis.

## **1.2 Apa itu data warehouse?**

Data warehouse adalah database yang berisi data dari beberapa system operasional yang terintegrasi, teragregasi dan terstruktur sehingga dapat digunakan untuk mendukung analisa dan proses pengambilan keputusan dalam bisnis.

### **1.2.1 Mengapa kita perlu Data Warehouse?**

Apakah semua informasi yang diperlukan untuk menjalankan bisnis tersedia pada saat dibutuhkan, dalam bentuk sebagaimana diperlukan dan cukup detail dengan tingkat kurasi yang tinggi untuk mendukung suatu keputusan ? Atau, apakah dua orang user yang bertemu pada suatu rapat (meeting) memiliki data laporan yang tidak sesuai ? Satu orang menganalisa penjualan untuk bulan Maret adalah \$500 juta sedang yang lain berkata ada \$524 juta Setelah dianalisa, terdapat perbedaan pada data yang digunakan untuk menghitung penjualan pada tiap laporan, dan anda perlu banyak waktu untuk mencoba memecahkan mengapa hal itu bisa terjadi dan bagaimana memecahkan permasalahan tersebut.

Apakah perusahaan memiliki beberapa system dengan fungsi yang sama – system inventory lama dan baru yang menghabiskan jutaan dollar untuk membangun system tersebut ? Apakah anda perlu mendapatkan data dari kedua system tersebut, mengkombinasikannya untuk mendapatkan sebuah laporan ? Seberapa bagus hal ini bisa dikerjakan ? Apakah user memahami perbedaan dari kedua system ini pada saat melakukan query secara online ? Jika ini membutuhkan pemrograman, apakah anda memiliki application backlog atau hasil pelaporan setiap eksekusi dari suatu aplikasi ?

Apakah anda memiliki data detail histori untuk tujuan analisa ? Berapa bulan dari data histori yang diperlukan untuk di-online-kan ? Apakah anda menyimpan level detail yang benar ? Apakah anda menyimpan semua data histori ? Apakah anda dapat menganalisa penjualan untuk tiap produk dari setiap daerah sebelum dan sesudah struktur pelaporan penjualan di-reorganisasi ? Data warehouse dibangun dengan tujuan memecahkan semua persoalan tersebut.

## 1.3 Tinjauan Historis

Pada tahun 1970-an aplikasi komersial pertama dibangun untuk operasi bisnis sehari-hari. Sistem ini dibangun pada computer mainframe, yang sangat mahal harganya. Hanya bisnis besar yang dapat menggunakan hardware seperti ini, programmer membuat program di perangkat tersebut, dan ada staf operasi yang melakukan pemeliharaan. Sistem-sistem ini berfokus pada bagaimana menyisipkan data baru dan membaca data secara berurutan menggunakan magnetic tapes.

Dengan perkembangan dari media penyimpanan, data dapat diakses secara langsung. Dampaknya mulai muncul system manajemen database pertama, yang mengorganisasi data secara hirarki atau dalam bentuk jaringan. Sistem database ini sangat kompleks. Programmer harus paham bagaimana cara data disimpan dalam media penyimpan dan melakukan navigasi data untuk mengenerate sebuah laporan. Programmer aplikasi menggunakan COBOL untuk membuat laporan. Dan diperlukan waktu beberapa hari atau minggu untuk menulis program pada setiap laporan yang dibuat. Laporan dicetak pada kertas computer kemudian didistribusikan secara manual pada user yang membutuhkan. Tidak akan pernah cukup programmer, selalu terjadi kekurangan programmer, sehingga selalu diperlukan application backlog. Pada saat data dapat diakses secara langsung, mulai dibangun system pemrosesan transaksi secara online yang pertama atau on-line transaction processing (OLTP).

Pada akhir tahun 70-an dan awal 1980, minicomputers seperti Digital's PDP-11 dan VAX 11/780 membuat harga hardware turun drastis. Mulanya data disimpan dalam database CODASYL yang kompleks, sangat sulit untuk diubah, dan sulit dipahami juga sulit untuk didesain. Semua itu berubah secara munculnya database relasional. Di tahun 1979, database Oracle merupakan software komersial pertama yang menggunakan system relasional.

Dengan menggunakan model relasional, data diorganisasi ke dalam bentuk table dengan baris dan kolom. Dengan menggunakan pointer untuk memelihara relasi diantara data, nilai data yang unik yang mengidentifikasi setiap baris data dalam table, seperti nomer customer. Model relasional cukup mudah dipahami, dan SQL merupakan bahasa yang digunakan untuk mengakses database. SQL tidak membutuhkan pemahaman tentang bagaimana data disimpan secara fisik. Dengan demikian proses membangun aplikasi menjadi lebih mudah dan manajemen database digunakan secara luas. Setelah system relasional di-rilis, banyak perusahaan mulai berlomba-lomba untuk mengembangkan produk yang menggunakan pengaksesan ke database relasional, dengan memberikan perintah query (ad hoc query), pembuatan laporan (reporting), dan untuk keperluan analisa (analysis tools).

Dengan kemunculan Personal Computer (PC), komputasi beralih dari mainframes ke system client/server. Aplikasi Oracle dikenalkan pertama kali di akhir tahun 1980-an. Perusahaan tidak

lagi membangun sendiri aplikasinya; tetapi juga mencuan software yang menyediakan fungsionalitas dasar dari beberapa vendor seperti Oracle, PeopleSoft, dan SAP.

Sebagai database relasional yang mapan di tahun 1980-an, system OLTP dibangun dengan menggunakan system relasional untuk melakukan otomasi aspek operasional dari bisnis. Aplikasi yang dibangun meliputi pemrosesan order, order entry, inventory, general ledger, dan akunting. Sistem OLTP melakukan otomasi proses dan merepresentasikan status system secara real time. Dalam aplikasi inventory, terdapat transaksi untuk menyisipkan item baru ke dalam inventory, menghapus items jika semua habis terjual, dan merubah jumlah (quantity), semua proses harus balance. Sejumlah data histori disimpan dalam system. Sehingga mudah untuk mendapatkan jumlah dari produk yang memiliki kode produk "114 45-222", atau kapan nomer order "45321" harus dikirim. Pada saat itu, vendor dari database relasional berfokus pada peningkatan performansi untuk aplikasi OLTP dan berkompetisi satu sama lain dengan menggunakan standart industry TPC-C benchmarks.

### **1.3.1 Era kemunculan data warehouse**

Pada saat system OLTP dibangun untuk efisiensi pengumpulan data, tantangan utamanya adalah seberapa bagus data ini bisa diinterpretasikan. Dan pada akhir tahun 1980-an dan awal tahun 1990-an, mulai dilakukan tinjauan bisnis secara menyeluruh, sehingga muncullah enterprise data warehouses untuk pertama kalinya (istilah datawarehouse dikenalkan pertama kali oleh Bill Inmon, "bapaknya data ware-housing"). Data diambil dari beberapa system operasional yang digunakan untuk menjalankan operasi bisnis sehari-hari untuk menyediakan tinjauan perusahaan secara menyeluruh.

Data warehouses dibangun untuk memberikan tinjauan bisnis dari waktu tertentu dan melihat trend-nya. Banyak keputusan yang diperlukan untuk melihat detail dari operasi sehari-hari yang membentuk suatu tinjauan bisnis yang lebih luas. Misal suatu query terhadap warehouse melibatkan tinjauan atas pelaporan jumlah penjualan produk dua tahun terakhir untuk tipe penjualan retail vs. penjualan via Internet. Query tersebut melibatkan berapa jumlah nilai yang berubah untuk suatu waktu tertentu, perubahan lain yang mengikutinya sehingga dapat ditemukan koneksi yang sebelumnya tidak diketahui.

Untuk membentuk tipe analisa seperti ini, data dalam warehouse perlu disimpan untuk suatu periode waktu tertentu, biasanya dari lima sampai 10 tahun.

#### **data warehouse melihat ke dalam data untuk menemukan informasi**

Dalam data warehouse, aktifitas utamanya adalah query, atau membaca data. Aktifitas perubahan (update) hanya terjadi pada saat data di-load ke dalam warehouse. Decision support systems (DSS), semisal Oracle Discoverer, menyediakan query yang interaktif, tampilan grafik, chart, dan pembuatan laporan. Oracle menambahkan tipe baru dari struktur pengaksesan yang dapat membantu meningkatkan performansi query dalam warehouse, semisal bitmap indexes. Kemudian materialized views ditambahkan pula sejak Oracle 8i untuk meningkat perfromansi pada saat melakukan agregasi dan summary pada data.

Oracle Express Server menjadi front end analisa pada data ware-house yang menyediakan on-line analytical processing (OLAP). Software OLAP digunakan untuk menganalisa data bisnis dalam hirarki top-down. Yang mengasumsikan query akan diproses secara iterative, dimana hasil dari jawaban suatu pertanyaan akan memungkinkan untuk menimbulkan banyak pertanyaan baru.

Tidak cukup hanya dengan mengetahui keuntungan yang diperoleh tahun ini, seorang analis juga perlu mengetahui keuntungan untuk setiap periode waktu tertentu untuk setiap produk untuk setiap area geografis tertentu. Karena melibatkan tiga hal (waktu, produk dan geografis) maka digunakan query tiga dimensi. Seorang analis perlu membandingkan penjualan bulan ini dengan bulan yang sama pada tahun sebelumnya untuk setiap toko. Kita bisa melakukan operasi drill down untuk melihat ke level yang lebih detail dalam suatu hirarki untuk mendapatkan toko di lokasi geografis mana mana yang paling banyak mendapatkan keuntungan.

### **Data warehouse membutuhkan desain database yang berbeda**

Data warehouses didesain untuk kita bisa melakukan query secara cepat. Informasi diturunkan dari data lain, dilakukan rolling up data untuk dijadikan ringkasan (summaries), dilakukan operasi drilling down untuk mendapatkan informasi lebih detail, atau melihat pola yang menarik atau melihat trend (kecenderungan).

Dalam system OLTP, teknik pembuatan ER (entity relationship) diagram digunakan untuk mendesain skema database. Tiap entity menjadi sebuah table dimana atribunya menjadi kolom, dan relasi direpresentasikan dengan melakukan join pada primary-key di suatu table dan foreign-key di table yang lain..

Desain ternormalisasi menyediakan performansi yang optimal untuk system OLTP, karena banyaknya transaksi yang diperlukan untuk meng-update data. Normalisasi memastikan table-table dibentuk secara benar dan menghindari terjadinya redundansi. Dengan hanya memiliki hanya satu kopi data, update anomalies dapat dihindari. Dengan meng-update data pada satu tempat, konsistensi tetap terjaga.

Untuk mengoptimalkan performansi warehouse, dimana aktifitas utama yang dilakukan adalah melakukan query atas data, maka diperlukan model data baru yang mendukung keperluan ini. Ralph Kimball, pelaku di bidang industry, pengarang buku *The Data Warehouse Toolkit* memperkenalkan skema model data baru yang disebut skema star (star schema), sebuah cara baru untuk mendesain database yang memfasilitasi pemrosesan OLAP. Dalam rangka mengoptimalkan performansi warehouse, digunakan teknik pemodelan dimensi.

Pendekatan dimensional pada pemodelan, mengorganisasi data ke dalam table fakta dan table-table dimensi. Data direpresentasikan dalam bentuk yang lebih mudah dipahami oleh user. User yang memerlukan laporan per-kuartal untuk tiap area dapat mengambil data berdasarkan dimensi waktu dan dimensi area. Dimensi-dimensi dalam skema star digunakan untuk melakukan analisa dan mengorganisasi data. Dengan menggunakan pemodelan dimensi, diperbolehkan adanya denormalisasi dan redundancy (sesuatu yang harus dihindari pada system OLTP). Pada Bab 2, akan dipelajari cara mendesain contoh kasus untuk perusahaan Easy Shopping Inc.

Desain logika ditampilkan dalam bentuk representasi fisik yang akan mengoptimalkan performansi dan manajemen. Dalam hal ini, Tabel, constraints, indexes, dan partisi didefinisikan.

Oracle menambahkan beberapa fitur untuk mendukung desain dimensional yang menggunakan skema star. Jadi selain membuat table, dimensi juga didefinisikan.

### **1.3.2 Data warehouses sebagai system terpisah**

Mungkin anda heran mengapa tidak kita gunakan saja system produksi operasional yang dipakai sehari-hari untuk data warehouse. Mengapa informasi yang diperlukan tidak diambilkan saja dari secara langsung untuk menjalankan bisnis secara efisien ? Mengapa kita perlu mengkopi data dari satu system ke system lain untuk membangun warehouse? Ini dikarenakan system operasional tidak berisi data histori (data time series – atau data yang mengandung dimensi waktu), sehingga data tidak available untuk dianalisa. Juga skema data tidak didesain untuk keperluan queri kecerdasan bisnis (business intelligence queries).

Query dalam warehouse mengakses data dalam jumlah yang sangat besar, sehingga membutuhkan memori yang cukup besar, CPU, dan sumber daya I/O yang besar pula. Menjalankan query untuk mendukung proses pengambilan keputusan membutuhkan komputasi yang besar dan lama sehingga dapat mempengaruhi performansi system operasional sehari-hari.

Seringkali system operasional dijalankan pada platform hardware yang berbeda dengan OS yang berbeda dan DBMS yang berbeda pula.

Database terdistribusi memungkinkan user untuk melakukan query dan global query optimizers akan mengalokasi data secara transparan, mengembalikan hasilnya pada user dalam waktu secepat mungkin sehingga user tidak menyadari bahwa data bertempat di mesin pada lokasi geografis yang berbeda. Tapi system seperti ini tidak akan pernah terwujud.

Pada kenyataannya, data perlu dipindah dari system operasional ke dalam system warehouse yang terpisah, yang menyimpan format secara umum untuk setiap sumber mesin yang berbeda-beda.

### **Data warehouse dibangun dari system operasional**

Membangun warehouse melibatkan ekstraksi data dari system operasional, seringkali data dikombinasikan dengan informasi tambahan dari pihak ketiga, mentransformasinya ke dalam format yang seragam dan di-load ke dalam database.

Sekali data di-load ke dalam warehouse, maka hampir tidak akan pernah diubah lagi. Hampir 80 persen waktu yang digunakan untuk membangun data warehouse digunakan untuk proses extraction/transformation/load (ETL): menempatkan data, menulis program untuk ekstraksi; memfilter, dan dan membersihkan data; mentransformasi data ke dalam skema pengkodean yang umum; kemudian me-load data ke dalam data warehouse.

Data operasional harus diekstraksi dari sistem operasional kemudian dikopi ke dalam *staging area*—yaitu lokasi temporer dimana data perlu dibersihkan, ditransformasi, disiapkan ke dalam warehouse.

Data dari beberapa sistem perlu ditransformasi ke dalam format umum yang disepakati dan digunakan dalam warehouse. Diperlukan pengetahuan tentang arti penyimpanan data dalam sistem operasional, semisal :

- Bisa jadi untuk suatu item yang sama memiliki dua nama yang berbeda, misal nama produk dengan kode "1234" disebut dengan "roti" sedangkan untuk kode yang sama "1234" di tempat lain disebut "kue"
- Setiap sistem mungkin menggunakan skema pengkodean yang berbeda. Misal kode produk di suatu tempat direpresentasikan dengan karakter yang dipisahkan dengan tanda '-' (xxx-xx-xxx) sedangkan di tempat lain untuk kode produk merupakan direpresentasikan dengan karakter yang dipisahkan dengan tanda spasi (xxx xx xxx). Jika dimasukkan dalam warehouse, format yang digunakan harus sama.
- Atribut tabel memiliki nama berbeda. Misal untuk atribut gender pada tabel customer suatu sistem menggunakan nilai "0" atau "1." Sedangkan di sistem lain, digunakan "M" atau "F."
- Setiap sistem menggunakan unit pengukuran (measure) yang tidak sama. Misal dollar digunakan di US, dan euro digunakan di Eropa. Data ini harus dikonversi ke dalam unit pengukuran yang sama jika dimasukkan ke dalam warehouse.

Dalam proses transformasi, nama harus diseragamkan, dan skema pengkodean yang dipakai harus seragam pula.

Jika data sekali ditransformasi, maka siap untuk di-load ke dalam ware-house. Seringkali area transformasi merupakan tempat yang terpisah.

Jika data yang telah ditransformasi disimpan dalam flat file, maka bisa ditransformasi dengan menggunakan FTP dan di-load dengan menggunakan utilitas SQL\*Loader atau tabel eksternal. Jika data yang ditransformasi berada dalam database Oracle, maka data dipindah antar tablespaces dari satu database ke database yang lain.

Oracle Warehouse Builder, adalah tool dalam the Oracle Internet Developer Suite, yang membantu melakukan otomatisasi ekstraksi, transformasi, transportasi, dan load ke dalam warehouse. Dibahas di Bab 5.

Data baru yang ditambahkan ke dalam warehouse secara periodik. Biasanya dilakukan load dalam pemrosesan batch di sore hari (setelah jam kantor) atau di waktu lain pada saat warehouse tidak digunakan untuk menganalisa.

Juga bisa digunakan data eksternal untuk ditambahkan ke dalam warehouse. Misal, informasi penting yang berisi data cuaca, demografi dan sosial ekonomi. Contoh penggunaannya :

- Menambahkan data cuaca , sehingga dapat diketahui produk apa yang meningkat atau menurun tingkat penjualannya sehubungan dengan keadaan cuaca tertentu.
- Menambahkan data demografi customer; memiliki strategi pemasaran, mengumpulkan customer target yang sekiranya bisa merespon suatu promosi penjualan.
- Untuk mengetahui tipe customers yang membeli tipe dari produk, untuk mengantisipasi meningkatnya permintaan akan barang. Data demografi dapat digunakan untuk membantu memilih lokasi untuk menempatkan lokasi toko yang baru.

### **1.3.3 Data mart**

Membangun warehouse merupakan hal yang sangat kompleks dan butuh waktu sekitar 18 bulan sampai tiga tahun. Dikarenakan warehouse berisi banyak area subyek dan lintas organisasi, juga berhubungan dengan situasi politik tertentu. Karena itu banyak proyek data warehousing yang gagal.

Juga telah banyak diperoleh keuntungan dari penggunaan warehouse, untuk memecahkan masalah bisnis. Data warehouses berisi banyak subyek yang menyediakan tinjauan perusahaan secara menyeluruh. Data marts adalah subyek spesifik atau aplikasi spesifik dari data warehouses yang berisi data untuk satu line bisnis tertentu semisal penjualan atau pemasaran. Perbedaan antara data mart dan data warehouse adalah pada ruang lingkup informasi yang disediakan. Ruang lingkup data mart lebih kecil, data didapatkan dari sumber yang lebih sedikit, dan waktu implementasinya juga lebih pendek.

#### **Independent data marts**

Data marts bisa bersifat dependent atau independent, tergantung pada sumber informasinya. Sumber informasi untuk data mart yang bersifat dependent adalah data warehouse itu sendiri. Sedangkan data mart bersifat independent jika tidak terdapat data warehouse, dan data diekstrak secara langsung dari system operasional.

Karenanya independent data marts dapat dibangun lebih cepat, dan mulai populer di pertengahan tahun 1990-an dimana tiap departemen dalam perusahaan membuat data mart untuk keperluannya sendiri. Tapi sayangnya, setelah pembuatan beberapa data marts, persoalan mulai muncul. Setiap data mart yang ada memiliki versi sendiri-sendiri. Dan muncul persoalan baru ketika dua report memiliki isi yang berbeda untuk suatu persoalan yang sama.

Salah satu alasan mengapa independent data marts dapat dibangun dengan cepat karena hanya data yang diperlukan oleh departemen tertentu yang perlu diidentifikasi dan dipahami. Bukan pemahaman terhadap keseluruhan perusahaan. Pembuatan independent data marts menghindari isu politik berkaitan dengan pembuatan aturan penamaan dan standart pengkodean.

Persoalan lain muncul dari kenyataan bahwa data marts yang dibangun secara independent berasal dari tim yang berbeda. Tim-tim ini seringkali memiliki hardware, software, dan tools yang berbeda pula. Tiap independent data mart mengambil data secara langsung dari system operasional. Jika perusahaan memiliki lima data marts yang berbeda, dimana masing-masing

memerlukan informasi customer, maka terdapat lima program terpisah yang dijalankan untuk mengekstrak data dari table customer yang pada system operasional.

Jika masing-masing data mart menjalankan proses pembersihan data (cleansing) dan transformasi data, maka kemungkinan cara yang dilakukan tidak akan sama. Data yang redundant dan tidak konsisten akan menyulitkan proses pengambilan keputusan.

Sebagaimana yang dibahas sebelumnya, terdapat perbedaan utama antara data dalam system operasional dan data dalam warehouse. Data operasional data terkini yang digunakan untuk manage operasi sehari-hari. Data dalam warehouse bersifat informasional, dan berisi data histori di waktu lampau.

Jika diperlukan informasi tentang status bisnis terkini untuk pembuatan keputusan taktis yang mendukung operasi sehari-hari, maka sebuah operational data store (ODS) dibangun sebagai bagian dari arsitektur manajemen informasi. ODS berorientasi pada subyek, data terintegrasi yang bersifat valid dan telah dibersihkan. Dan digunakan untuk mendukung query operasional dan pembuatan report. Sebagai contoh, organisasi pelayanan customer perlu mengakses informasi account balances dan billing.

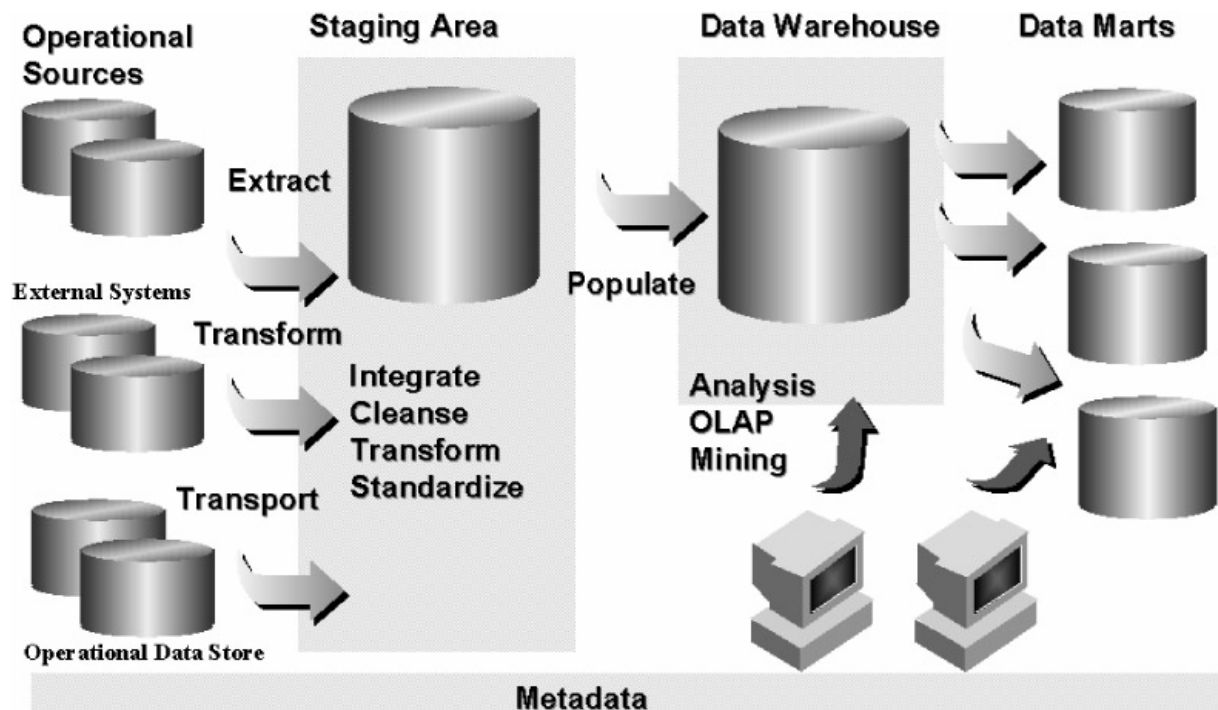
ODS dapat diupdate hampir mendekati proses real time, karena mencerminkan keadaan saat sekarang. ODS dapat menjadi sumber bagi ware-house atau data marts.

### **Secara bertahap membangun data warehouse yang berasal dari dependent data marts**

Daripada membangun warehouse yang keseluruhan mencerminkan perusahaan, fungsi dari warehouse biasa dibangun mulai dari pendekatan secara bertahap.

Gambar [1.1](#) menampilkan arsitektur yang digunakan saat ini. Dimana data diekstrak dari system OTLP dan sumber eksternal, di-load ke dalam operational data stores (ODS) dan enterprise data warehouses, kemudian di-load ke dalam dependent data marts.





Gambar 1.1: Enterprise data warehouse dengan dependent data marts.

Membangun data warehouse seperti membangun software. Kita tidak bisa mengerjakan keseluruhan hanya dalam satu rilis, dan tidak akan bisa diantisipasi semua hal yang mungkin. Akan lebih baik jika dibangun arsitektur secara keseluruhan, membangun framework dengan komponen yang memungkinkan warehouse dibangun secara bertahap. Membatasi ruang lingkup, dan merencanakan perbaikan setiap tiga sampai enam bulan berikutnya.

### Reporting, query, dan tools untuk analisa

Di tahun 1995 Larry Ellison, CEO dari Oracle, pertama memperkenalkan visi dari jaringan computer : merupakan perangkat yang kecil, tidak mahal yang membuatnya mudah untuk menjalankan aplikasi yang mengakses informasi melalui Internet.

Di tahun 1998 Oracle merilis aplikasi yang dapat dijalankan di Web, dan menghentikan pengembangan aplikasi yang digunakan di lingkungan client/server. Oracle8i, merupakan Internet database, dirilis dengan beberapa fitur baru yang didesain untuk mendukung aplikasi berbasis Internet.

Kemampuan untuk mempublish report di Web dapat membuat informasi selalu tersedia dimanapun tanpa perlu diinstall terlebih dahulu. Ini membuat pegawai, rekanan, dan customer secara realtime dapat mengakses informasi penting. Kita tidak perlu berada di kantor untuk dapat menampilkan sebuah report. Dengan menempatkan informasi di Web ini berarti lokasi kantor bisa dimanapun.

## 1.4 Oracle 9i

Oracle merilis Oracle 9i Release 2 di tahun 2002, produk teknologinya yang utama yaitu : Oracle 9i Database, Oracle 9i Application Server, 9iAS, dan Oracle 9i Developer Suite. Komponen yang sebelumnya terpisah sekarang terintegrasi dalam suite.

Oracle secara dramatis merilis software untuk application server, Oracle 9iAS yang melengkapi teknologi untuk membangun portals, menganalisa trafik Web traffic, dan membuat aplikasi wireless. Oracle 9iAS Discoverer workbooks dan Oracle 9iAS reports dipublish sebagai portlets. Oracle 9iAS clickstream intelligence dapat memberikan laporan efektifitas dari Web dengan menggunakan Web server logs.

Dengan dilengkapi Oracle 9i Developer Suite, tools dapat membuat aplikasi Business Intelligence (Kecerdasan Bisnis) berbasis Internet. JDeveloper menyediakan lingkungan pengembangan secara cepat untuk membuat crosstabs, graphs, dan elemen lain yang digunakan untuk analisa dan presentasi.

### 1.4.1 Fitur-fitur Data warehousing dalam 9i Database

Beberapa fitur ditambahkan ke dalam Oracle 9i Database untuk meningkatkan performansi, manajemen dan skalabiliti dari data warehouse. Fitur-fitur tersebut adalah :

- Bitmap join indexes untuk meningkatkan performansi pada saat join table.
- Metode partisi baru yang memungkinkan DBA untuk menampilkan daftar nilai yang disimpan dalam tiap partisi.
- Peningkatan pada proses ETL (Extract-Transform Load):
  - Merubah data capture yang menyediakan mekanisme untuk mengidentifikasi perubahan baris dalam Oracle database
  - Tabel eksternal yang memungkinkan penyediaan transformasi pada saat meload data ke dalam database
  - Perintah SQL baru yaitu Merge yang membentuk operasi yang berguna pada saat meload data dimensi. Jika baris sudah ada, maka dimodifikasi, jika belum ada maka disisipkan sebagai data baru.
  - Fungsi table yang menyediakan eksekusi parallel dari transformasi menggunakan perintah PL/SQL
- Materialized views dapat dilakukan secara bertahap (cepat) direfresh setelah perintah DML individual, setelah direct path loads

Software Business intelligence (BI) menyediakan pengetahuan yang relevan terhadap data untuk proses pelaporan dan analisa. Meliputi : tools untuk end-user query dan pelaporan, OLAP tools, data mining tools, dan executive information systems (EIS), yang menyediakan operasi untuk melakukan drill down, navigasi dan exception reporting. Dalam Oracle 9i, beberapa fungsi business intelligence telah ditambahkan ke dalam database.

### 1.4.2 Bertemunya Oracle 9i Database dan teknologi express server

OLAP pertama kali didefinisikan oleh Dr. E. F. Codd, penemu database relasional. Codd menyatakan bahwa database relasional tidak dimaksudkan untuk menyediakan sintesa data, analisa dan konsolidasi – fungsi yang telah didefinisikan sebagai analisa multi dimensi. Setelah beberapa tahun, database analitikal yang terpisah seperti Oracle Express menyediakan fungsionalitas yang tidak tersedia dalam database relasional.

Oracle menambahkan banyak fitur yang memfasilitasi OLAP queries, dan Oracle 9i Release 2, dimungkinkan untuk menggunakan Oracle server secara langsung untuk OLAP. Perintah SQL dikembangkan untuk menyediakan fungsi analitik, semisal ranking, moving window aggregates, period-over-period comparisons, ratio to report, fungsi statistik, inverse percentiles, hypothetical rank dan distributions, histograms, dan first/last aggregates. Beberapa level agregasi dapat dihitung dengan menggunakan cube, rollup, dan grouping sets. Fungsi-fungsi ini memungkinkan OLAP dapat diekspresikan tanpa memerlukan self-join yang kompleks dan memungkinkan optimizer untuk memilih execution plan yang lebih baik.

### **1.4.3 Fungsionalitas Data mining dalam Oracle 9i Database**

Oracle 9i Data Mining, merupakan bagian dari Enterprise Edition, yang memasukkan fungsionalitas data mining ke dalam database untuk membuat fungsi klasifikasi (classifications), prediksi (predictions), dan asosiasi (associations).

Data mining adalah bagian dari proses penemuan pengetahuan. Dengan menggunakan teknik statistik, data yang berukuran sangat besar dapat ditransformasi untuk mendapatkan informasi yang berguna. Data perlu digali untuk mendapatkan sesuatu yang berharga di dalamnya yaitu informasi dan kalau lebih digali lagi didapatkan pengetahuan.

Data mining mengekstrak informasi baru atau pengetahuan dari dalam data. Ini akan memungkinkan kita untuk mengetahui potongan informasi yang tidak diketahui sebelumnya sehingga dapat digunakan untuk mendukung keputusan.

Data berukuran sangat besar akan dibaca, dilihat kesamaannya satu sama lain, dikelompokkan untuk mendeteksi adanya pola yang menarik dan trend.

Tool untuk OLAP dan DSS melihat relasi asosiasi dalam struktur data. Ini semua dinyatakan oleh batasan (constraints) dan dimensi (dimensions). Data mining mendeteksi adanya relasi yang berhubungan dengan isi data yang tidak diketahui sebelumnya, semisal produk yang memiliki kecenderungan untuk dibeli secara bersamaan, permasalahan seperti ini dikenal sebagai market-basket analysis. Pada saat menganalisa data dengan fungsi waktu, data dapat digunakan untuk mendeteksi suatu pola. Beberapa aplikasi yang dikerjakan dengan data mining diantaranya adalah : customer retention, fraud detection, dan pola pembelian customer. Data digali untuk mencari pola menarik yang berguna untuk merancang strategi pemasaran.

Tool OLAP memungkinkan kita untuk mencari jawaban dari pertanyaan : Apakah penjualan lampu lava meingkat di bulan November jika dibandingkan dengan penjualan di tahun sebelumnya ? Tool Data mining dapat membantu mengidentifikasi jawaban dari pertanyaan seperti : Faktor apa yang menentukan penjualan lampu lava ?

Dengan menggunakan tool OLAP, analis memulai pertanyaan atau hipotesa dan melakukan queri pada warehouse untuk membuktikan teori tersebut. Dengan menggunakan tool data mining, pekerjaan yang sebelumnya dilakukan oleh analis berpindah ke computer. Tool data mining dapat menggunakan berbagai macam teknik untuk memecahkan sejumlah persoalan yang berbeda.

Oracle 9iAS Personalization, adalah aplikasi yang dapat memberikan rekomendasi secara real-time, dengan menggunakan Oracle 9i Database yang memiliki fungsionalitas Data Mining untuk secara dinamis memberikan rekomendasi bagi pengunjung Web dan customer yang teregistrasi. Oracle 9iAS Personalization dapat menjawab pertanyaan semisal : Yang mana item (barang) yang paling disukai ? Apakah orang yang membeli barang tersebut juga ingin membeli barang yang lain ? Tipe pertanyaan personal seperti ini dapat dikirim di Amazon.com atau di banyak Web site yang lain.

## **1.5 Membangun sebuah data warehouse memiliki banyak tantangan**

Meskipun data warehousing telah matang sebagai teknologi yang digunakan dalam beberapa tahun terakhir, akan tetapi pengembang data warehouse menghadapi banyak tantangan baru sejalan dengan lingkungan bisnis dan teknologi yang senantiasa berubah. Database harus dibangun untuk dapat menampilkan performansinya yang terbaik dan keandalannya (reliability) dapat harus dapat mengantisipasi skala bisnis (scalability) yang semakin tumbuh.

### **1.5.1 Memanage- warehouse**

Sejalan dengan pertumbuhan ukuran data warehouses yang sampai terabytes, dengan kebutuhan ketersediaan yang lebih tinggi, maka prosedur backup dan recovery harus direncanakan dan baik isi data maupun penggunaannya dalam warehouse harus di-manage.

Dalam system OLTP, aplikasi dirancang untuk memproses data transaksi secepat mungkin. Dalam data warehouse, performansi dirancang untuk memproses sebanyak mungkin variable query..

Warehouse perlu di-tunning untuk mendapatkan performansi yang lebih baik. Siapa yang menggunakan data ? Level summary mana yang dilihat user ? Data mana yang sedang digunakan ? Apakah struktur data telah terdefinisi dengan benar ? Apakah telah diindex pada kolom yang tepat ? Dapatkan table summary digunakan oleh query ? Jika banyak query yang men-join satu table dengan table lain, apakah perlu dilakukan denormalisasi ? Informasi workload akan membantu untuk menentukan index yang tepat yang perlu ditambahkan, table-tabel yang perlu dikombinasikan (denormalisasi) dan bagaimana suatu summary seharusnya dibuat.

Pada dasarnya tidak perlu menyimpan semua detail data untuk dapat diakses secara langsung. Data dapat dibersihkan tanpa perlu dikopi atau data dapat diarsip atau dipindah ke media penyimpanan, dapat dapat diambil jika diperlukan.

### 1.5.2 Perlunya metadata

Metadata adalah data yang menjelaskan tentang data lain atau operasi yang berlaku pada data. Metadata dapat digunakan untuk tujuan bisnis maupun teknis. Sebagaimana aliran data dari system operasional dimasukkan ke dalam warehouse, diekstrak, ditransformasi, dan di-summary. Metadata secara teknik diperlukan untuk menjelaskan proses ini dan diperlukan pada operasi "drill down" untuk mencari level detail yang benar. Metadata secara bisnis membantu user untuk menentukan data apa yang tersedia dalam warehouse atau data mart dan bagaimana data dapat diakses. Metadata menyediakan integrasi dan keseragaman data pada lintas organisasi. Metadata merupakan tempat dimana departemen yang berbeda mendeskripsikan penggunaan dari istilah "produk". Metadata disimpan dalam repository, yang mana merupakan kumpulan table dalam database Oracle. Metadata dapat di-share oleh user atau tool.

Di tahun 2000 Object Management Group (OMG) mempublish spesifikasi "Common Warehouse Metamodel (CWM)," yang mendefinisikan format metadata untuk semua produk data warehouse dan business intelligence. Spesifikasi tersebut dikembangkan berdasarkan kerjasama dari beberapa perusahaan, termasuk Oracle dan IBM.

### 1.5.3 Volume data yang meningkat

Salah satu isu atau topic yang mendominasi saat ini adalah terjadinya ledakan dari ukuran data. Data warehouses merupakan database berukuran sangat besar. Ada beberapa alasan terjadinya ledakan data, diantaranya :

- Dengan semakin murah harga hardware, maka secara ekonomi memungkinkan untuk menyimpan semakin banyak data histori. Data pembelian barang oleh customer di supermarket, tidak hanya menyimpan total pembeliannya tapi juga dapat disimpan barang apa saja yang dibeli.
- Dunia bisnis menyimpan lebih banyak data untuk waktu yang lama.
- Data disimpan berulang kali untuk keperluan yang berbeda. Indexes dan materialized views dibuat untuk meningkatkan performansi query, tapi untuk bisa mengakses strukturnya membutuhkan tambahan ruang penyimpanan sejalan dengan semakin besarnya ukuran database.
- Data yang tidak terstruktur dapat diintegrasikan dengan aplikasi business intelligence tradisional. Penyimpanan data multimedia membutuhkan ukuran yang besar. Untuk satu jam data video membutuhkan sekitar 1 GB. Untuk menyimpan satu menit data audio membutuhkan sedikitnya 1MB. Data image berkisar antara 20 KB sampai 60 MB tergantung pada tipe dan kualitas gambar yang disimpan.
- Dokumen dapat menggunakan format metadata berbasis XML dan disimpan dalam Oracle 9i.

Akan tetapi perlu diingat bahwa data warehouse seharusnya dipandang tidak hanya sebagai repository untuk mengarsip data; karena ini bukan satu-satunya tujuan.

### 1.5.4 Ketersediaan (availability) yang tinggi

Kepastian akan ketersediaan data warehouse menjadi hal yang sangat kritis dan penting.

Oracle 9i Database didesain untuk mengeliminasi kebutuhan akan melakukan down time pada system disebabkan karena : kerusakan pada system (system failure), kerusakan penyimpanan (storage failure), kerusakan site (site failure), atau kesalahan manusia (human error). Jika server di-down, aplikasi anda akan tetap jalan. Karena real application clusters (RAC) membuat aplikasi tetap scalable dan memiliki ketersediaan yang tinggi. Sebuah database tunggal dapat dijalankan pada kumpulan server yang ter-cluster bersama-sama. Jika ada server ditambahkan ke dalam cluster, aplikasi dapat meningkatkan throughput-nya (scalability-nya lebih besar), tanpa perlu modifikasi.

Data Guard dapat digunakan untuk me-maintain kopi dari isi datawarehouse secara konsisten, dapat dipastikan bahwa operasi dapat berlanjut dengan interupsi yang minimal dikarenakan kerusakan, human error, atau data corruption.

### **1.5.5 Lebih banyak users/performansi lebih baik**

Kemampuan untuk mem-publish reports di Web membuat informasi selalu tersedia setiap saat untuk orang banyak. Dengan semakin tersedianya data warehouses dan tool untuk business intelligence, maka pengguna juga semakin banyak. Kebutuhan akan performansi yang baik menjadi lebih penting daripada sebelumnya, apalagi queri yang terlibat juga semakin kompleks.

### **1.5.6 Tipe-tipe aplikasi yang baru**

Data warehouses dapat digunakan untuk mendukung beberapa kegiatan bisnis, seperti customer relationship management (CRM) dan supply chain management. CRM membantu untuk member daya tarik pada customer baru dan mengembangkan loyalitas customer, hal yang penting disini adalah mempertahankan loyalitas customer. Data ware-house dapat berisi informasi tentang data customers yang seringkali terintegrasi dengan data penjualan, pemasaran dan aplikasi layanan customer (customer care applications).

## **1.6 Data warehouse di masa depan**

Akan kemana lagi Data warehouse dapat dikembangkan ? Ada beberapa alternatif untuk pengembangan Data warehouse di masa depan, antara lain seperti yang akan dibahas berikut ini.

### **1.6.1 Real-time data warehouses**

Data warehouse dapat mendukung analisa secara analisa dan proses pengambilan keputusan secara real time. Data warehouse tidak perlu di-update secara periodic dalam system batch, karena jika suatu transaksi sudah komit di system OLTP, maka pada saat itu juga akan langsung tersedia dalam data warehouse, sehingga dimungkinkan kemampuannya untuk mendukung pengambilan keputusan secara real time.

Hal ini memungkinkan missal perusahaan kartu kredit untuk mendeteksi dan menghentikan terjadinya penyalahgunaan (fraud detection), atau perusahaan transportasi untuk menyusun ulang rutanya setelah terjadi kecelakaan di suatu tempat, dan penjual on-line untuk berkomunikasi secara langsung dengan menawarkan sesuatu yang special pada pembeli potensial yang sedang surfing di Web.

### **1.6.2 Bukan merupakan system terpisah**

Suatu hari nanti hanya diperlukan satu database tunggal untuk OLTP dan data warehousing. Tidak perlu lagi memisahkan database untuk OLTP, ODS, data warehouse, dan data marts. Hal ini akan menghemat tenaga, waktu dan biaya untuk memindah data berukuran besar untuk proses ekstraksi, transformasi dan load (ETL - extraction, transformation, loading) dan juga replikasi lintas database.

Baik OLTP, pengambilan keputusan dan pelaporan dapat dijalankan pada instance Oracle yang sama dengan menggunakan RAC. Dengan Oracle 9i, tidak perlu lagi memisahkan engine untuk relasional, OLAP, data mining, atau ETL. Ini akan menyederhanakan operasi dan manajemen dari infrastruktur data warehouse.

Tentu saja, sebelumnya perlu diintegrasikan data dari banyak sumber, sampai semuanya siap untuk disimpan dalam satu database Oracle.

### **1.6.3 Ringkasan**

Kita telah beranjak dari era mainframe di tahun 1970-an, ke era minicomputers di tahun 1980-an, ke era client/server di tahun 1990-an. Dan saat ini, komputasi Internet telah mengalami perubahan yang paling signifikan, dikarenakan semakin tingginya layanan di bidang bisnis.

Pada bab ini kita telah mempelajari latar belakang dari kemunculan data warehouses dan data marts, memahami secara garis besar fungsionalitas dari Oracle 9i, beberapa tantangan yang dihadapi oleh pengembang warehouse di masa depan. Dan tibalah saatnya, sekarang kita akan menggunakan semua teknologi ini untuk membangun dan mengakses data warehouse.