

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Pustaka

Bisnis di seluruh dunia menghadapi tantangan untuk tetap kompetitif, pentingnya BI untuk kelangsungan hidup tidak boleh diremehkan. Semakin meningkatnya perusahaan menjadi lebih otomatis, *data-driven*, dan *real-time*, arsitektur BI berkembang untuk mendukung pengambilan keputusan (Dayal et.al., 2009). Cavalcanti (2005) mengevaluasi sejauh mana hubungan antara tingkat aktivitas BI dan persepsi kesuksesan bisnis. Temuan penelitian ini menunjukkan hubungan positif antara penerapan BI dan kesuksesan bisnis di perusahaan-perusahaan besar Brasil.

Bisnis menyadari bahwa dalam lingkungan bisnis yang sangat kompetitif yang serba cepat dan selalu berubah, kuantitas kompetitif utama adalah seberapa cepat mereka merespon dan beradaptasi dengan perubahan. BI memungkinkan mereka untuk menggunakan informasi yang dikumpulkan dengan cepat dan terus-menerus merespon perubahan (Ranjan, 2009). Hasil analisis Popovic & Jaklic (2010) menunjukkan bahwa sistem BI benar-benar memiliki dampak positif pada kualitas informasi. Kualitas konten informasi lebih penting untuk membuat keputusan bisnis yang lebih baik dan memberikan nilai lebih tinggi pada sistem BI.

Dalam beberapa tahun terakhir, pasar BI telah mengalami pertumbuhan yang sangat tinggi, berbagai vendor terus melaporkan laba yang cukup besar. Survei menunjukkan bahwa menurut kebanyakan CIO, BI adalah aplikasi yang paling penting untuk dibeli. Hasil survei tentang pengeluaran CIO juga menemukan bahwa area dengan prioritas belanja paling banyak adalah BI (Yeoh et.al., 2008).

Pada tahun 2005 pasar dunia untuk solusi BI naik sekitar 11,5 persen, mencapai nilai 5,7 miliar dolar. Implementasi paling banyak dihabiskan oleh perusahaan-perusahaan Amerika (52,9 persen dari jumlah di atas), pada posisi kedua adalah negara-negara dari kawasan EMEA (terutama Eropa - 35,8 persen),

pada pengguna tempat ketiga ditempatkan dari Asia dan Pasifik (11 , 3 persen) (Oleskow et.al., 2006). Platform BI menjadi lebih luas dan canggih, dan meningkatkan jumlah perusahaan yang mengintegrasikan solusi BI, seperti Amazon.com, Netflix, Walmart, Proctor and Gamble, CapitalOne, Harrah, Marriott, dan Red Sox semua bersaing pada BI dan analisis prediktif (Brannon, 2010).

Venter & Tustin (2009) menyajikan temuan-temuan yang paling menonjol dari penelitian BI baru-baru ini, yang merupakan salah satu dari beberapa studi yang telah dilakukan di Afrika Selatan pada abad ke-21. Hasilnya adalah kurangnya perusahaan yang memanfaatkan peluang BI dan mengkoordinasikan fungsi BI secara efektif. Untuk mengatasi masalah informasi di perusahaan manufaktur kecil dan menengah, dibentuk platform berorientasi manufaktur kecil dan menengah yang didasarkan pada teknologi BI ASP (Shen & Ding, 2008 ; Ponelis & Britz, 2011).

Terdapat beberapa penelitian yang membahas penerapan BI pada produsen dan penyedia layanan pengelolaan rantai pasok secara lebih efisien dan menguntungkan (Smietana, 2010). Penelitian yang ekstensif telah dilakukan pada aplikasi untuk optimasi rantai pasokan operasional, diimplementasikannya BI memungkinkan untuk berbagi, mengintegrasikan, dan menganalisis data RFID. Berdasarkan hasil penelitian ini, model data multidimensi telah diterapkan dalam lingkungan OLAP (Baars et.al., 2008). Penelitian lain menyajikan pengembangan sebuah proyek BI pada industri yang sesungguhnya, dalam sebuah perusahaan yang menggunakan sistem ERP menggunakan aplikasi Oracle (Lupu et.al., 2007).

Penelitian Santosa et.al. (2011) adalah sebuah aplikasi OLAP yang menyajikan pola karakteristik konsumen dalam melakukan pembelian terhadap suatu produk pada *e-commerce* melalui sebuah halaman web. Cunningham et.al. (2006) merancang *data warehouse* untuk mendukung analisis CRM. Studi menunjukkan bahwa model perdana ini dapat digunakan untuk menganalisis berbagai analisis profitabilitas seperti analisis profitabilitas pelanggan, analisis profitabilitas pasar, analisis profitabilitas produk, dan analisis profitabilitas saluran.

Penelitian dalam bidang kesehatan juga dilakukan, Demigha (2010) mengembangkan sistem *data warehouse* di radiologi-senology untuk membantu membantu skrining kanker payudara dalam diagnosis, pendidikan, dan penelitian. Budhi et.al. (2010) merancang *star schema* untuk mendukung menganalisis historis *track record* pada RSUD Dr Soetomo dengan menggunakan Oracle 9i. Atzmueller et.al. (2009) memanfaatkan *data warehouse* yang berisi data dari sistem cerdas untuk dievaluasi dan dari sumber eksternal lain pada sebuah studi kasus dalam domain medis sonografi.

Penelitian pada organisasi pemerintah juga telah dilakukan, sebuah *framework* bagi BI dalam *e-governance* disajikan untuk lebih memperkuat sistem *e-governance*. (Coman, 2009). Iraqi Health and Social Care Organization menggunakan *data warehouse* untuk mengidentifikasi kebutuhan, merencanakan proyek dan mengumpulkan dana untuk korban sisa perang (Hassin, 2008). Sementara yang lebih dekat ke bidang pendidikan adalah penelitian untuk mengidentifikasi perubahan-perubahan di perpustakaan karena penerapan teknologi dan bagaimana teknologi *data warehouse* dapat membantu untuk menemukan pengetahuan dan meningkatkan pelayanan (Wikramanayake & Goonetillake, 2006).

Organisasi semakin membutuhkan penggunaan BI berbasis web dengan teknologi pengolahan *on-line* analitis untuk memahami dan untuk mendapatkan wawasan kompetitif dalam volume data yang besar. Keuntungan kunci dari alat berbasis web bila dibandingkan dengan alat-alat tradisional yang berbasis *batch* atau klien server termasuk kemudahan penggunaan, akses universal di seluruh platform teknologi informasi, dan respon dalam hitungan menit dan umpan balik berdasarkan data dinamis dan *real-time* (Heinrichs & Lim, 2003; Kursan & Mihić, 2010).

Web telah menjadi platform pilihan untuk pengiriman aplikasi bisnis untuk perusahaan skala besar. Akhirnya, teknologi web membuat *data warehouse* dan sistem pendukung keputusan lebih *user-friendly* (Grabova et.al., 2010). Sebuah BI berbasis web dibangun untuk memungkinkan para pengambil keputusan di universitas untuk mengevaluasi berbagai program dan departemen akademik dari

waktu ke waktu. Sistem ini berfungsi sebagai titik akses ke *data warehouse* universitas (Rhodes & Kincheloe, 2000).

BI dengan *data warehouse* menghasilkan laporan yang dapat dipercaya dalam waktu yang singkat bagi manajemen tingkat atas perguruan tinggi. Tak pelak lagi sudah saatnya *data warehouse* harus diimplementasikan pada perguruan tinggi (Warnars, 2008). Sudah ada beberapa penerapan BI yang digunakan pada proses pembelajaran sebagai proses utama pendidikan. Gombiro et.al (2008) memperkenalkan konsep *data warehouse* untuk membantu dalam berbagi materi pembelajaran di Zimbabwe, sedangkan Lyman et.al. (2002) mengevaluasi penggunaan *data warehouse* akademik untuk informasi pasien di sekolah medis. Perancangan *data warehouse* untuk penerimaan mahasiswa baru dilakukan oleh Cardoso et.al. (2003) di Universitas Teknik Lisbon. Akintola et.al. (2011) membuat prototipe *data warehouse* untuk pendaftaran kelas menggunakan Microsoft SQL Server Analysis Services tanpa laporan. Binh (2008) mengusulkan desain *framework* sistem penasihat akademik dengan membangun prototipe integrasi data saja.

Penerapan BI menggunakan laporan dan *dashboard* telah dikembangkan, namun penyajian laporan dalam penelitian-penelitian ini belum berbasis web. Wiak et.al. (2010) mengimplementasikan *data warehouse* untuk meningkatkan pendidikan dan mendiagnosa permasalahan di Technical University of Łódź menggunakan Microsoft SQL Server 2008 dan laporannya menggunakan Microsoft Excel 2007. Alnoukari (2009) membantu pembuatan laporan untuk top manajemen universitas untuk kinerja mahasiswa dan kinerja dosen menggunakan Oracle Warehouse Builder.

Beberapa penelitian berikut telah lebih berkembang dengan memanfaatkan internet untuk penyajian laporan berbasis web. Prototipe *data warehouse* untuk menganalisa performa e-learning pada universitas dilakukan oleh Muntean et.al. (2011) dan Solodovnikova & Niedríte (2005) menggunakan Oracle. Vinnik & Scholl (2007) membuat prototipe *data warehouse* untuk manajemen beban kerja akademik di universitas. Dimokas et.al. (2008) membuat prototipe *data warehouse* untuk membantu menganalisis data pendidikan menggunakan Microsoft SQL

Server 2005 dan pelaporannya menggunakan Microsoft SharePoint Server 2007. *Data warehouse* untuk mendukung keputusan manajemen dalam pengukuran kinerja universitas dibangun dengan Microsoft SQL Server 2005 (Păunică et.al., 2010) dan Oracle (Muntean et.al, 2010). Ta'a et.al. (2008) membuat prototipe *data warehouse* di area subyek kemahasiswaan menggunakan SAS.

Menurut Wilarso (2008), perkembangan pemanfaatan teknologi *data warehouse* di perguruan tinggi Indonesia jauh tertinggal dengan di beberapa negara maju seperti Amerika. Telah ada penelitian yang mengembangkan BI di Indonesia untuk pengukuran kinerja proses belajar mengajar di jurusan Teknik Informatika Universitas Kristen Petra oleh Handoyo dan Rostianingsih (2004). Namun, informasi yang didapatkan disajikan dalam bentuk *pivot table* pada Microsoft Excel saja. Rezalini et.al. (2010) mengimplementasikan BI untuk menilai kondisi akademik mengenai data mahasiswa dan data IPD (Indeks Prestasi Dosen) menggunakan Pentaho Business Intelligent Suite di jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember. Berdasarkan segi manfaatnya, proyek *data warehouse* pada perguruan tinggi layak untuk diimplementasikan menurut Warnars (2009), yang meneliti Simple ROI untuk justifikasi investasi proyek *data warehouse* pada perguruan tinggi swasta.

Penerapan *real time* BI sedikit disinggung di beberapa penelitian, pendekatan *real-time* BI dalam analisis rantai pasokan akan memperoleh efisiensi operasional dan KPI yang lebih baik untuk setiap organisasi dalam SCM (Sahay & Ranjan, 2008). Continental Airlines digunakan sebagai studi kasus yang sangat sukses menerapkan *real-time* BI yang menggambarkan sebuah aplikasi yang menggambarkan penggunaan di Continental secara *real-time* (Watson et.al., 2006; Anderson-Lehman et.al., 2004).

Penelitian BI di universitas yang telah dibahas di atas, masih menggunakan ETL tradisional. Penerapan *real-time* BI sedikit disinggung di beberapa penelitian, penelitian mengenai CDC sebagai teknik penerapan *real-time* BI masih sedikit ditemukan. Diantaranya, desain algoritma CDC pada aplikasi perbankan pada penelitian Jain et.al. (2012), serta penelitian yang membahas

tentang teori dibalik ETL dan evolusi ETL termasuk di dalamnya CDC (Kakish & Kraft, 2012).

B. Landasan Teori

1. *Business Intelligence*

Pada sub bab ini akan dijabarkan tentang pengertian, latar belakang, dan klasifikasi BI.

a. Pengertian *Business Intelligence*

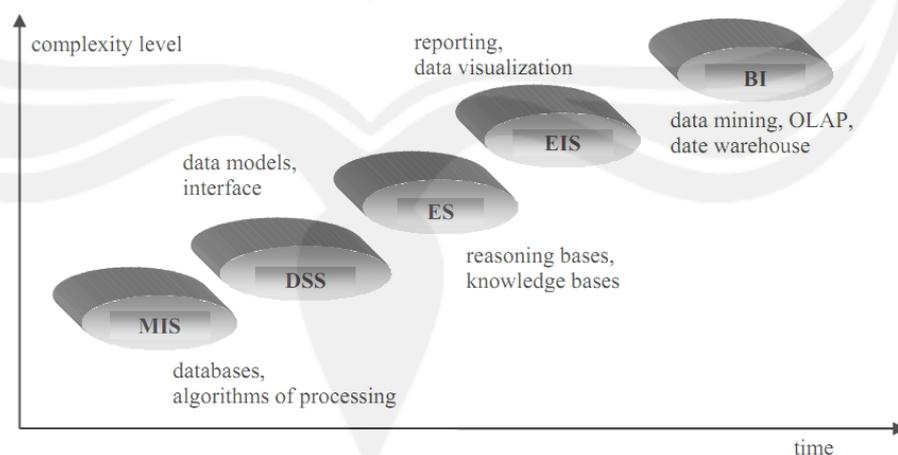
Berikut ini adalah pengertian BI dari beberapa ahli:

- 1) Menurut Turban et al. (2007), sebuah istilah umum yang meliputi alat, arsitektur, basis data, *data warehouse*, manajemen kinerja, metodologi, dan sebagainya, yang semuanya terintegrasi ke dalam paket perangkat lunak terpadu.
- 2) Menurut Moss & Atre (2003), arsitektur dan koleksi operasional terpadu serta aplikasi pendukung keputusan dan basis data yang menyediakan akses mudah komunitas bisnis ke data bisnis.
- 3) Chang (2006), akurat, tepat waktu, data kritis, informasi dan pengetahuan yang mendukung pengambilan keputusan strategis dan operasional dan penilaian risiko di lingkungan bisnis yang tidak pasti dan dinamis. Sumber data, informasi, dan pengetahuan baik internal keorganisasian yang dikumpulkan maupun eksternal yang dipasok oleh mitra, pelanggan atau pihak ketiga sebagai akibat dari pilihan mereka sendiri.
- 4) Menurut Gangadharan & Swami (2004), hasil analisis mendalam dari data bisnis yang terperinci, termasuk teknologi basis data dan aplikasi, serta praktik analisis.
- 5) Moss & Hoberman (2004), proses, teknologi, dan peralatan yang diperlukan untuk mengubah data menjadi informasi, informasi menjadi pengetahuan dan pengetahuan menjadi rencana yang mendorong tindakan bisnis yang menguntungkan. BI meliputi *data warehouse*, alat-alat analisis bisnis dan konten/manajemen pengetahuan.

b. Latar Belakang *Business Intelligence*

Untuk waktu yang lama *Management Information Systems* (MIS) telah mendukung organisasi dalam tugas-tugas yang berbeda. Namun, saat ini banyak sistem telah mengalami penyusutan yang signifikan. Sampai saat ini MIS yang ada (yaitu MIS, DSS, ES, EIS) tidak selalu memenuhi harapan para pengambil keputusan, seperti: membuat keputusan di bawah tekanan waktu, pemantauan kompetisi, memiliki beberapa informasi pada organisasi yang mencakup sudut pandang yang berbeda, dan melakukan analisis data yang konstan dan mempertimbangkan berbagai varian yang berbeda dari kinerja organisasi (Olszak & Ziembra, 2007).

MIS yang ada tidak menangani integrasi dari data yang berbeda, tersebar dan heterogen, mereka tidak dapat menafsirkan data tersebut dalam konteks yang luas secara efektif dan mereka tidak mampu menemukan data baru yang cukup saling ketergantungan (Olszak & Ziembra, 2007). Dalam rangka untuk dapat bereaksi dengan cepat terhadap perubahan yang terjadi di pasar, organisasi membutuhkan MIS yang akan memungkinkan untuk melakukan analisis penyebab dan pengaruh yang berbeda dari organisasi itu sendiri dan lingkungan mereka (Power, 2001).



Gambar 2.1 Perkembangan *Business Intelligence*

Sistem BI menyediakan sebuah solusi yang dapat menghadapi kebutuhan organisasi. Tugas utama yang akan dihadapi oleh sistem BI meliputi eksplorasi cerdas, integrasi, agregasi, dan analisis multidimensi dari data yang berasal dari

berbagai sumber informasi. Sistem standar BI menggabungkan data dari sistem informasi internal dari suatu organisasi dan mengintegrasikan data yang berasal dari lingkungan tertentu, misalnya statistik, keuangan, investasi, dan basis data lainnya. Sistem seperti ini dimaksudkan untuk memberikan informasi yang memadai dan dapat diandalkan mengenai berbagai aspek kegiatan organisasi (Olszak & Ziemba, 2007).

c. Klasifikasi *Business Intelligence*

Menurut White (2006), BI dikelompokkan menjadi 3 (tiga) kategori, di antaranya:

- 1) *Strategic BI* : digunakan untuk mengelola rencana bisnis jangka panjang dan tujuan. Eksekutif dan manajer senior menggunakan metrik kinerja bisnis tingkat tinggi (disebut *key performance indicators*, atau KPI) yang diproduksi oleh *strategic BI* untuk melacak seberapa baik bisnis yang dilakukan terhadap tujuan jangka panjang bisnis seperti pertumbuhan pangsa pasar, pengurangan biaya, dan peningkatan pendapatan.
- 2) *Tactical BI* : mendukung inisiatif bisnis yang akan diluncurkan (kampanye pemasaran, produk baru, misalnya) untuk membantu menyelaraskan kinerja bisnis yang sebenarnya dengan kinerja yang direncanakan, analisis *tactical BI* digunakan oleh manajer senior, analis bisnis, dan manajer *line-of-business* (LOB) untuk mengukur dan mengoptimalkan kinerja dari inisiatif. *Tactical BI* menganalisis operasi bisnis selama periode hari, minggu, atau bulan.
- 3) *Operational BI* : Deteksi penipuan, manajemen risiko, segmentasi pelanggan, manajemen jaringan, dan manajemen persediaan adalah contoh dari proses operasional yang dapat diperbaiki dengan menggunakan BI operasional.

Tabel 2.1 Klasifikasi *Business Intelligence*

	<i>Strategic BI</i>	<i>Tactical BI</i>	<i>Operational BI</i>
Fokus bisnis	Mencapai tujuan bisnis jangka panjang	Mengatur inisiatif taktis untuk mencapai tujuan strategis	Mengatur dan mengoptimalkan operasi bisnis sehari-hari
Pengguna utama	Eksekutif dan analis bisnis	Manajer senior, analis bisnis, manajer LOB	Manajer LOB, pengguna LOB, dan sistem operasional
Time frame	bulanan ke tahunan	harian ke mingguan ke bulanan	sehari-hari
Data	<i>Historical metrics</i> (KPI)	<i>Historical metrics</i>	<i>Right-time metrics</i>

2. *Online Transactional Processing* (OLTP) dan *Online Analytical Processing* (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.

a. OLTP

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.

b. OLAP

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. *Data Warehouse*

Dalam subbab ini akan dijelaskan mengenai pengertian *data warehouse*, karakteristik *data warehouse*, dan komponen *data warehouse*.

a. Pengertian *Data Warehouse*

Menurut pelopor konsep dan istilah *data warehouse*, William Inmon, definisi dari *data warehouse* adalah:

Sebuah koleksi data yang berorientasi subjek, terintegrasi, non-volatile, dan time-variant dalam rangka mendukung keputusan-keputusan manajemen (Inmon, 2005).

Beberapa definisi lain dari beberapa sumber-sumber resmi sebagai berikut:

Data warehouse merupakan basis data yang bersifat analisis dan read only yang digunakan sebagai fondasi dari sistem penunjang keputusan (Poe,1998).

Data warehousing merupakan basis data relasional yang didesain lebih kepada query dan analisa dari pada proses transaksi, biasanya mengandung history data dari proses transaksi dan bisa juga data dari sumber lainnya. Data warehousing memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung/konsolidasi data dari berbagai macam sumber (Lane,2003).

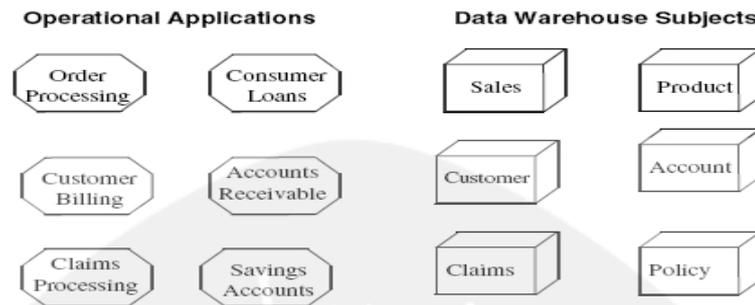
Dari definisi-definisi yang dijelaskan tadi, dapat disimpulkan *data warehouse* adalah basis data yang saling bereaksi yang dapat digunakan untuk *query* dan analisis, bersifat orientasi subjek, terintegrasi, *time variant*, tidak berubah yang digunakan untuk membantu para pengambil keputusan.

b. Karakteristik *Data Warehouse*

Ponniah (2010) menyebutkan karakteristik dari *data warehouse* adalah sebagai berikut:

1) Berorientasi Subjek

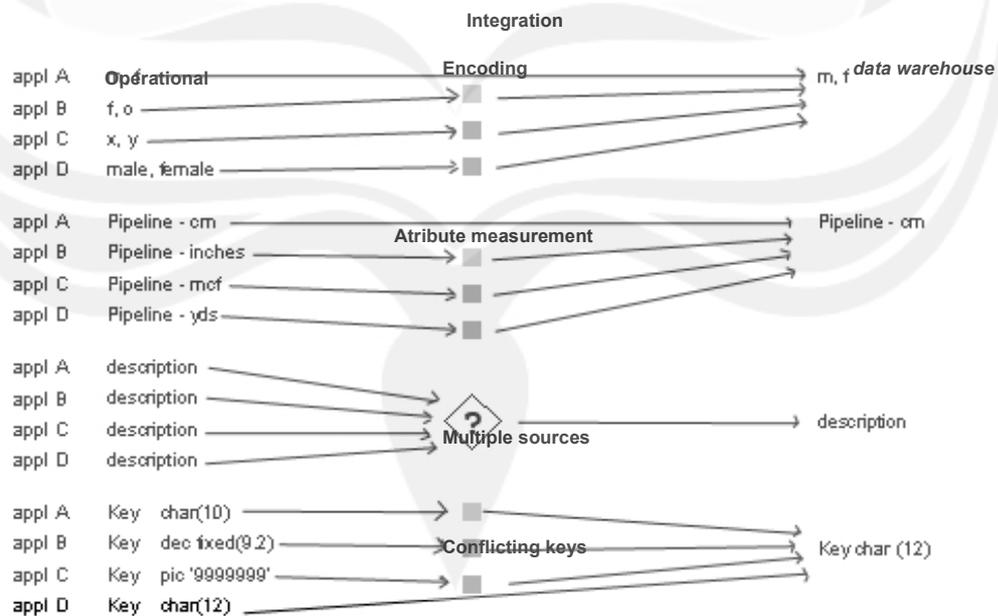
Data warehouse didesain untuk menganalisa data berdasarkan subjek-subjek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu. *Data warehouse* diorganisasikan disekitar subjek-subjek utama dari perusahaan (konsumen, produk dan penjualan) dan tidak diorganisasikan pada area-area aplikasi utama (*customer invoicing, stock control* dan *product sales*).



Gambar 2.2 Perbedaan *Data Warehouse* dan Basis Data Operasional (Ponniah, 2010)

2) Terintegrasi

Sumber data yang ada dalam *data warehouse* tidak hanya berasal dari data operasional (*internal source*) tetapi juga berasal dari data di luar sistem (*external source*). *Data warehouse* dapat menyimpan data-data yang berasal dari sumber-sumber yang terpisah ke dalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara seperti konsisten dalam penamaan variabel, konsisten dalam ukuran variabel, konsisten dalam struktur pengkodean, dan konsisten dalam atribut fisik dari data.



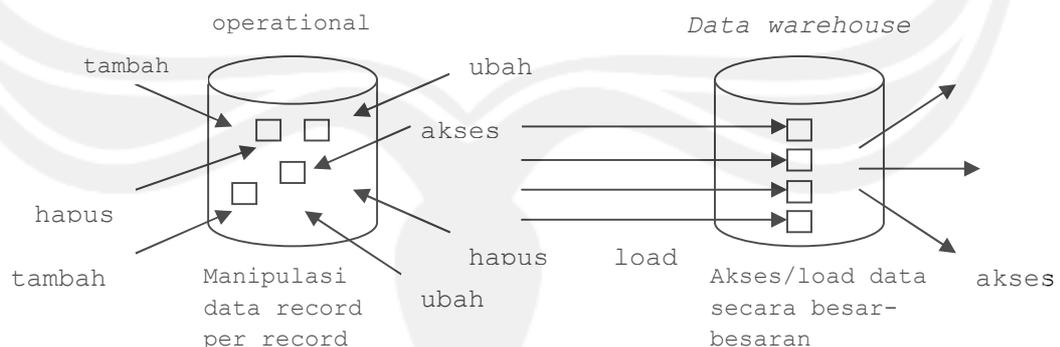
Gambar 2.3 Masalah Integrasi (Inmon, 2005)

3) *Time-variant*

Sistem operasional mengandung data yang bernilai sekarang sedangkan data dalam *data warehouse* mengandung data tidak hanya data terkini tetapi juga data masa lampau yang akan digunakan dalam analisis dan pengambilan keputusan. Waktu adalah dimensi penting yang harus didukung oleh semua *data warehouse*. Data untuk analisis dari berbagai sumber berisi berbagai nilai waktu, misalkan harian, mingguan, dan bulanan.

4) *Non-volatile*

Data dalam basis data operasional akan secara berkala atau periodik dipindahkan ke dalam *data warehouse* sesuai dengan jadwal yang sudah ditentukan. Misal perhari, perminggu, perbulan, dan lain sebagainya. Data yang baru selalu ditambahkan sebagai suplemen bagi basis data itu sendiri daripada sebagai sebuah perubahan. Basis data tersebut secara kontinyu menyerap data baru ini, kemudian secara inkremen disatukan dengan data sebelumnya. Basis data operasional bisa dibaca, diperbarui, dan dihapus. Tetapi pada *data warehouse* hanya ada dua kegiatan memanipulasi data yaitu *loading* data (mengambil data) dan akses data (mengakses *data warehouse* seperti melakukan *query* atau menampilkan laporan yang dibutuhkan, tidak ada kegiatan *updating* data).



Gambar 2.4 Masalah *Nonvolatility* (Inmon, 2005)

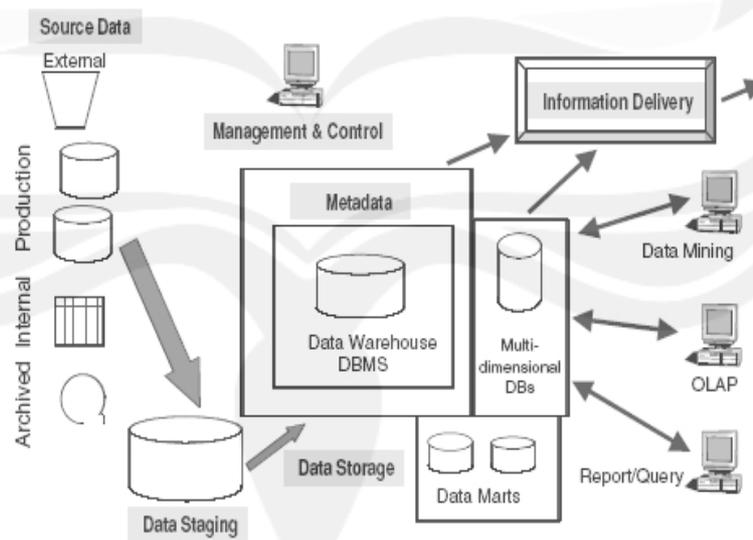
5) *Granularity*

Pada sistem operasional data dibuat secara *real-time* sehingga untuk mendapatkan informasi langsung dilakukan proses *query*. Pada *data warehouse* pada menganalisis harus memperhatikan detail per level misalkan perhari, ringkasan

perbulan, ringkasan per-tiga-bulan. Granularitas menunjuk pada level perincian atau peringkasan yang ada pada unit-unit data dalam *data warehouse*. Semakin banyak detail yang ada, maka semakin rendah level granularitas. Semakin sedikit detail yang ada, maka semakin tinggi level granularitas. Semakin tinggi level granularitas maka *query* yang dapat ditangani oleh *data warehouse* semakin terbatas. Semakin rendah level granularitas maka *query* yang dapat ditangani oleh *data warehouse* semakin banyak dan jawaban *query* yang diperolehpun semakin detail.

c. Komponen *Data Warehouse*

Pada subbab ini, akan dibahas tentang komponen-komponen *data warehouse* menurut 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.



Gambar 2.5 Komponen *Data Warehouse* (Ponniah, 2010)

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-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-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-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 industri 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:

(1) *Format Revisions*

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

(2) *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.

(3) *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.

(4) *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*.

(5) *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.

(6) *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.

(7) *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.

(8) *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.

(9) *Key Restructing*

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

(10) *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:

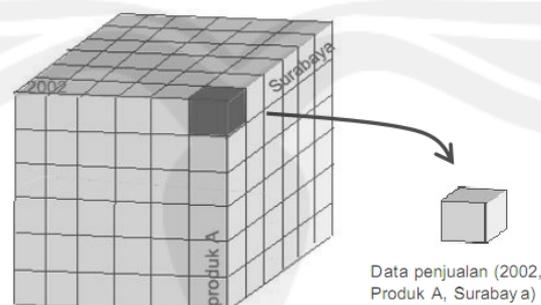
- a) Pengguna baru: menggunakan *data warehouse* tanpa pelatihan dan karena itu memerlukan *template-template* laporan dan *query-query* yang telah ditetapkan sebelumnya.
- b) Pengguna tidak tetap: memerlukan informasi hanya sesekali, tidak secara teratur. Pengguna tipe ini juga memerlukan paket informasi 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.

4. Model Data 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 2.6 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. Di dalam 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*

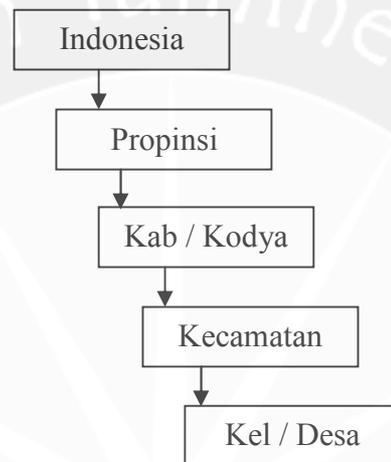
Nilai *measure* terletak pada tabel fakta. *Measure* juga cerminan dari fakta dan juga mengandung data yang akan dianalisa seperti contoh pada gambar 2.7. OLAP memerlukan informasi kolom bertipe numerik yang akan dijadikan *measure*.

Sales Fact Table	
time_key	} measure
item_key	
location_key	
unit_sold	
avg_sales	
dollars_sold	

Gambar 2.7 *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 2.8.



Gambar 2.8 Hirarki Dimensi Daerah

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 di atasnya, 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

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*.

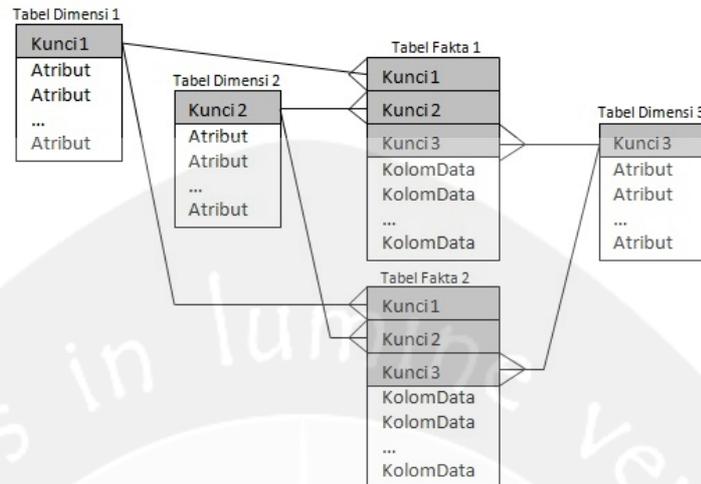
g. *Cube*

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).

a. *Star Schema*

Star schema terdiri dari satu atau lebih tabel fakta dan satu atau lebih tabel dimensi. Tabel fakta merupakan pusat dari *star schema*, karena fungsinya sebagai pengikat dari tabel-tabel dimensi yang terletak di sekelilingnya. Hubungan antara tabel-tabel tersebut menggunakan *foreign key*. Gambar 2.9 menunjukkan pemodelan basis data multidimensi dengan *star schema*.



Gambar 2.9 Star Schema

b. Snowflake Schema

Snowflake schema merupakan pengembangan dari *star schema*. Perbedaannya terletak pada normalisasi data dan jumlah tabel. Pada *snowflake schema* tabel dimensi dinormalisasi secara sebagian atau keseluruhan untuk mengurangi nilai duplikat pada tabel. Dengan kata lain satu atau lebih tabel dimensi tidak bergabung secara langsung kepada tabel fakta tapi pada tabel dimensi lainnya. Gambar 2.10 menunjukkan pemodelan data menggunakan *snowflake schema*.



Gambar 2.10 Snowflake Schema

5. Real-time Business Intelligence

a. Pengertian Real-time Business Intelligence

Arti *real-time* BI terutama tergantung pada pemahaman apa arti '*real-time*' untuk bisnis. '*Real time*' dapat berarti: kebutuhan untuk mendapatkan latensi nol dalam proses, proses memiliki akses ke informasi kapan saja diperlukan, proses memberikan informasi bilamana diperlukan oleh manajemen, memiliki kemampuan untuk mendapatkan ukuran kinerja kunci yang berhubungan dengan

situasi pada titik saat ini, tidak hanya untuk beberapa situasi historis (Azvine et.al., 2006).

Berdasarkan penjelasan ini, *real-time* BI menyediakan fungsi yang sama seperti BI tradisional, tetapi beroperasi pada data yang diambil dari sumber data operasional dengan latensi nol, dan menyediakan sarana untuk menyebarkan aksi ke proses bisnis secara *real-time*. Secara khusus, *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).

Dalam *real-time* BI data dianalisis segera setelah memasuki organisasi. Dalam konteks ini, *real-time* berarti memberikan informasi dalam rentang milidetik ke beberapa detik setelah kejadian bisnis. Sebagai contoh, lini manajer bisnis dapat memantau tingkat persediaan dalam rangka untuk memastikan bahwa kampanye pemasaran *on-line* tidak akan gagal karena situasi stok tidak tersedia. Sebuah sistem *real-time* BI didasarkan pada *data warehouse real-time*, *data warehouse* meningkatkan siklus *refresh* untuk memperbarui data lebih sering. Sistem *data warehouse real-time* dapat mencapai hingga mendekati *update* data *real-time* (Sandu, 2008).

b. *Real-time Business Intelligence* dengan *Change Data Capture*

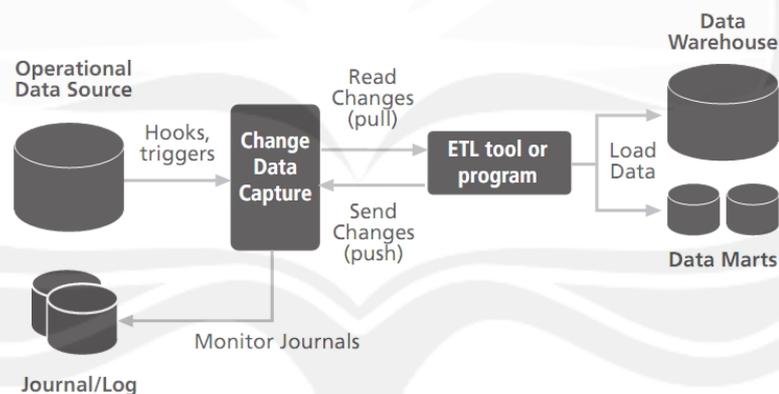
Change Data Capture (CDC) adalah pendekatan inovasi untuk integrasi data, berdasarkan identifikasi, menangkap, dan mengirimkan perubahan yang dibuat oleh data sumber. Dengan memproses hanya perubahan data saja, CDC membuat proses integrasi data lebih efisien dan mengurangi biaya dengan mengurangi latensi (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 *event-delivery* 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 *real-time*.

Diagram berikut menunjukkan CDC pada proses ETL secara umum:

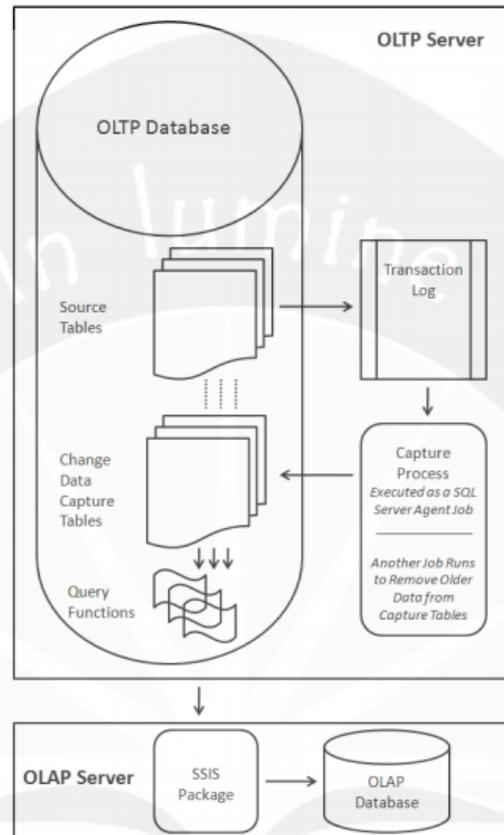


Gambar 2.11 Integrasi CDC Pada ETL (Attachmate Corp, 2005)

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 2.12 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.
- 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 2.12, 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 mengizinkan 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.

6. SQL Server 2008

Microsoft menyediakan sebuah BI komprehensif untuk integrasi data, *data warehouse*, analisis, dan pelaporan, termasuk alat yang kuat dan intuitif yang dapat digunakan oleh pengguna akhir untuk mengakses dan menganalisis informasi bisnis. Microsoft SQL Server 2008 merupakan platform data lengkap yang memungkinkan untuk menyatukan penyimpanan dan akses untuk semua data

di seluruh perusahaan, serta membangun dan mengelola solusi BI canggih (Microsoft, 2008).

Spesifik teknologi dari SQL Server 2008 yang membentuk dasar dari BI yang kuat dijelaskan dalam tabel berikut (Microsoft, 2008).

Tabel 2.2 Komponen SQL Server 2008

Komponen	Deskripsi
SQL Server Database Engine	Mesin penyimpanan data berperforma tinggi untuk volume data yang sangat besar menjadikannya pilihan ideal untuk mengkonsolidasikan data bisnis dari seluruh perusahaan ke dalam <i>data warehouse</i> pusat untuk analisis dan pelaporan
SQL Server Integration Services	Sebuah platform yang komprehensif untuk operasi ETL yang memungkinkan sinkronisasi <i>data warehouse</i> dengan data dari sumber data yang berbeda yang digunakan oleh aplikasi bisnis di seluruh organisasi
SQL Server Analysis Services	Menyediakan mesin analitis untuk solusi OLAP, termasuk agregasi <i>measure</i> bisnis pada lebih dari beberapa dimensi dan KPI, dan untuk solusi <i>data mining</i> yang menggunakan algoritma khusus untuk mengidentifikasi pola, tren, dan asosiasi data bisnis
SQL Server Reporting Services	Sebuah solusi pelaporan yang luas yang memudahkan untuk membuat, menerbitkan, dan mendistribusikan laporan bisnis yang terperinci baik dalam perusahaan dan luar perusahaan

SQL Server 2008 memberikan platform BI yang komprehensif, melalui integrasi yang mendalam dengan perangkat produktivitas, seperti sistem Microsoft Office

2007, yang dapat memberdayakan karyawan di seluruh perusahaan untuk menggunakan platform ini dan mengubah wawasan bisnis ke dalam tindakan yang efektif.

7. Report Portal

ReportPortal adalah aplikasi klien yang menyediakan akses ke berbagai sumber data melalui internet. Aplikasi ini tidak memerlukan *software* lain pada sisi klien selain Internet Explorer 5.5 atau lebih tinggi. Aplikasi ini ditulis dengan menggunakan teknologi Microsoft: JavaScript AJAX, ASP.NET, VB.NET, dan Microsoft SQL Server, juga menggunakan Microsoft XML untuk Analisis untuk akses data OLAP (www.reportportal.com).

ReportPortal menyediakan akses ke data Microsoft SQL Server Analysis Services OLAP menggunakan laporan sebagai berikut (www.reportportal.com):

- a. OLAP Report: sebuah tabel pivot yang memungkinkan pengguna mendesain laporan secara *online* dengan *drag and drop measure*, dimensi dan hirarki.
- b. Microsoft OWC (Office Web Components) Report: sebuah tabel pivot yang memungkinkan pengguna mendesain laporan secara *online* dengan *drag and drop* ukuran, dimensi dan tingkatan. Laporan ini mensyaratkan Microsoft Office Web Components diinstal pada mesin klien.
- c. Analysis Services 2005 KPI Report: Laporan ini memungkinkan pengguna melihat Key Performance Indicators yang dibuat dalam Analysis Services 2005.
- d. Data Mining Report: Laporan ini memungkinkan pengguna menelusuri model *data mining* yang dibuat dalam Analysis Services.

Report Portal menyediakan laporan visualisasi data untuk data OLAP menggunakan laporan sebagai berikut (www.reportportal.com):

- a. Pie-Chart Tree Report: memberikan kemampuan untuk memvisualisasikan berbagai dimensi pada satu halaman.
- b. Bar-Chart Tree Report: memberikan kemampuan untuk memvisualisasikan berbagai dimensi pada satu halaman. Metode visualisasi informasi ini

memungkinkan lebih dari satu ukuran dapat dipilih sebagai lawan untuk satu ukuran tunggal dari Pie-Chart Tree.

- c. Tree Map Report: memberikan kemampuan memvisualisasikan banyak data pada satu halaman. Pengguna dapat memilih dua tingkatan (detail dan pengelompokkan) dan dua ukuran (ukuran dan warna).

Report Portal menyediakan akses ke *Relational Data Sources* menggunakan laporan sebagai berikut (www.reportportal.com):

- a. ROLAP Report: sebuah tabel pivot yang memungkinkan pengguna mendesain laporan secara *online* dengan *drag and drop measure*, dimensi dan hirarki.
- b. SQL Report: sebuah laporan tabel yang dirancang secara *online* menggunakan SQL Builder.
- c. Crystal Report: laporan ini memungkinkan tampilan Crystal Reports.
- d. Reporting Services Report: laporan ini dapat merancang dan menampilkan dari Microsoft Reporting Services Report.

Report Portal juga menyediakan laporan lainnya untuk membantu mengorganisasikan informasi:

- a. Dashboard Report: laporan dashboard menempatkan beberapa laporan pada satu halaman web.
- b. Key Performance Indicators (KPI) Report: laporan KPI memberikan kemampuan untuk merancang dan menampilkan laporan KPI.
- c. File upload: setiap jenis *file* dapat di-*upload* ke dalam aplikasi. *File* tersebut disimpan dalam basis data.
- d. Hyperlink: sebuah *hyperlink* dapat dibuat ke halaman eksternal. Memiliki semua sifat-sifat laporan dan dapat berupa *web folder*.