

BAB II

LANDASAN TEORI

2.1 Inventori

Menurut Tersine (1984,3) istilah inventori dapat digunakan untuk mengartikan beberapa hal yang berbeda:

1. Stock dalam persediaan yang ada pada waktu yang ditentukan (aset nyata yang dapat dilihat, ditimbang, dan dihitung).
2. Daftar barang pada hak milik.
3. (Sebagai kata kerja) kegiatan menimbang dan menghitung banyaknya item yang ada dalam persediaan.
4. (Untuk catatan finansial dan akuntansi) nilai dari stok barang yang dimiliki oleh sebuah organisasi pada waktu tertentu.

2.1.1 Tipe-tipe Inventori

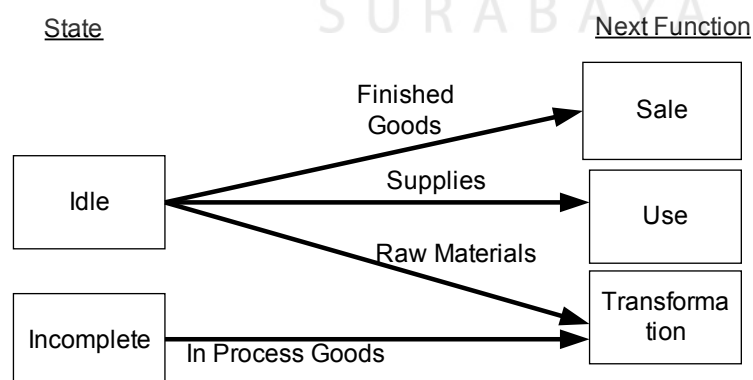
Inventori biasanya terdiri atas persediaan-persediaan, bahan baku, barang yang sedang diproses. Persediaan adalah materi inventori yang dikonsumsi di dalam penggunaan yang normal dari suatu organisasi yang bukan bagian dari produk akhir. Persediaan yang biasanya terdapat dalam inventori adalah pensil-pensil, catatan/kertas-catatan/kertas, bola lampu, media penyimpanan, alat pemotong, dan barang-barang untuk pemeliharaan fasilitas. (Perlengkapan pabrik disebut dengan MRO (*Maintenance, Repair, and Operating supplies*)).

Bahan-bahan mentah adalah materi yang dibeli dari para supplier untuk digunakan sebagai input untuk proses produksi. Bahan-bahan tersebut akan diolah

untuk menjadi barang jadi. Bahan baku untuk pabrik mebel adalah kayu olahan, bahan pewarna, lem, sekrup, pernis, paku, cat, dan sebagainya.

Dalam proses pembuatan mebel, barang adalah produk akhir yang diselesaikan secara parsial dan masih dalam proses produksi. Barang - barang tersebut merepresentasikan akumulasi dari pekerjaan yang diselesaikan secara parsial, dan antrian barang material yang menunggu proses yang lebih lanjut. Barang jadi adalah produk akhir, yang telah siap untuk penjualan, pendistribusian, atau penyimpanan.

Tugas inventori pada setiap kategori-kategori ini tergantung pada entity yang ada pada penjelasan ini. Hal ini terjadi karena produk jadi dari satu entity dapat menjadi bahan baku bagi yang lain. Contohnya, pabrik kulkas menganggap tabung tembaga sebagai bahan baku, tetapi perusahaan yang memproduksi tabung ini menganggapnya sebagai barang jadi. Customer untuk persediaan barang jadi kemungkinan adalah konsumen akhir, sebuah organisasi retail, distributor grosir, atau pabrik lain.



Gambar 2.1 Alur dan Fungsi Masing-masing Tipe Barang.

2.1.2 Fungsi Inventori

Berdasarkan kegunaannya, inventori dapat dikelompokkan kedalam kategori berikut :

- A. Working Stock
- B. Safety Stock
- C. Anticipation Stock
- D. Pipeline Stock
- E. Decoupling Stock
- F. Psycic Stock

Working Stock (dikenal sebagai siklus atau tumpukan persediaan) adalah inventori yang diperoleh dan ditahan dahulu sebelum timbul kebutuhan, sehingga pemesanan dapat dilakukan dalam jumlah yang besar, dibanding jika pemesanan dilakukan berdasarkan banyaknya kebutuhan. Perhitungan ukuran tumpukan persediaan bertujuan meminimalkan biaya pemesanan dan penahanan barang, diskon karena kuantitas barang, atau biaya untuk kualitas pengangkutan yang baik. Secara umum, jumlah rata-rata inventori dalam persediaan yang dihasilkan dari banyaknya tumpukan merupakan Working Stock dari suatu organisasi

SafetyStock (sering disebut dengan buffer atau stok fluktuasi) adalah inventori yang disimpan sebagai cadangan untuk perlindungan terhadap ketidakpastian permintaan dan penawaran. Safety stock membagi rata ke jumlah stok yang ada ditangan selama siklus penambahan sebagai perlindungan terhadap kekurangan stok.

Anticipation Stock (juga dikenal sebagai stok musiman atau stok stabilisasi) adalah inventori yang dibentuk untuk mengatasi permintaan pada

puncak musim, keperluan yang tidak menentu (program promosi, pemogokan, atau penutupan karena liburan), atau defisiensi dalam kapasitas produksi. Anticipation stock disediakan atau diproduksi sebelum adanya kebutuhan dan dikosongkan selama puncak periode permintaan untuk menjaga nilai tingkat produksi dan menstabilkan tenaga kerja.

Pipeline Stock (biasanya diartikan sebagai stok transit atau pekerjaan-dalam-proses) adalah inventori yang ditransit agar waktu yang dibutuhkan untuk menerima material pada akhir masukan, waktu yang dibutuhkan selama proses produksi, dan mengirimkan barang pada akhir output. Secara eksternal, pipeline stock adalah inventori yang berada di truk-truk, kapal-kapal, dan motorel atau dalam suatu saluran secara harafiah. Secara internal, pipeline stock adalah inventori yang sedang diproses, yang sedang menunggu untuk diproses, dan yang sedang dipindah.

Decoupling Stock adalah inventori yang dikumpulkan dari aktivitas-aktivitas yang saling bergantung, atau langkah-langkah untuk mengurangi persyaratan untuk operasi yang di sinkronisasi secara penuh. Decoupling stock mengisolasi salah satu bagian sistem dari bagian berikutnya agar masing-masing bagian dapat beroperasi dengan lebih bebas (tidak bergantung dengan bagian lain). Jadi, decoupling stock bertindak sebagai pemberian minyak pelumas untuk sistem penyediaan-produksi-distribusi yang melindungi terhadap friksi (pergeseran) yang berlebihan.

Psychic Stock adalah tampilan inventori eceran yang ditampilkan untuk menstimulasi permintaan dan bertindak sebagai "sales yang tidak berbicara". Psychic stock meningkatkan kesempatan bagi sebuah item untuk dilihat dan

kemudian dipertimbangkan untuk dibeli. Rak yang penuh meningkatkan penjualan dengan cara mengekspose sebanyak mungkin stok kepada para pelanggan dan menciptakan kesan jarak penglihatan produk yang lebih baik bagi para pelanggan. Rak yang kekurangan stok seperti halnya jika persediaan habis dapat mendorong kearah hilangnya penjualan dan hilangnya pelanggan. Jika kategori stok yang lain mendukung operasi dengan biaya rendah, psychic stock adalah kategori yang membangkitkan pendapatan. Kategori ini lebih concern kepada meningkatnya pendapatan melalui penciptaan permintaan daripada meminimuman biaya yang sifatnya berbasis persediaan.

2.1.3 Biaya Inventori

Sasaran dari manajemen inventori adalah sampai memiliki jumlah bahan-bahan yang sesuai pada tempat dan waktu yang tepat, dan dengan biaya rendah. Biaya-biaya inventori dihubungkan dengan pengoperasian suatu sistem inventori dan akibat dari tindakan atau tidak adanya tindakan pada pihak manajemen di dalam menetapkan sistem. Mereka adalah parameter-parameter ekonomi yang dasar untuk model keputusan inventori manapun dan yang kebanyakan relevan untuk sebagian besar sistem diperinci sebagai berikut:

- A. Purchase cost.
- B. Order/setup cost.
- C. Holding cost.
- D. Stockout cost.

Perlu diingat bahwa untuk item inventori tertentu, hanya elemen biaya yang bersifat incremental (keluar dari kantung) adalah bersangkutan di dalam analisa.

Biaya pembelian (P) dari suatu item adalah harga pembelian unit tersebut jika didapat dari sumber eksternal, atau biaya produksi unit tersebut jika diproduksi secara internal. Biaya unit sebaiknya selalu dianggap sebagai ongkos item selama ia ditempatkan di dalam inventori. Untuk item-item yang dibeli, biaya pembelian adalah harga beli ditambah biaya pengangkutan. Untuk barang yang dibuat oleh pabrik, biaya per unit termasuk juga biaya tenaga kerja langsung, bahan-bahan langsung, dan biaya overhead pabrik. Biaya pembelian dimodifikasi untuk tingkatan-tingkatan kuantitas yang berbeda ketika supplier menawarkan potongan karena kuantitas.

Biaya order/setup (C) dimulai dari biaya saat mengeluarkan Purchase Order (permintaan pembelian) ke supplier luar atau dari biaya-biaya setup produksi internal. Biaya ini pada umumnya diasumsikan berselang-seling secara langsung dengan banyaknya order(pesanan) atau setup yang ditempatkan dan sama sekali tidak diasumsikan dengan banyaknya order/ pesanan. Biaya pesanan meliputi item2 berikut, seperti membuat daftar permintaan, menganalisa vendor-vendor, menulis purchase order (pesanan pembelian), menerima bahan-bahan, memeriksa bahan-bahan, menindak lanjuti order/pesanan dan melakukan pengolahan yang diperlukan untuk melengkapi transaksi. Biaya Setup terdiri dari biaya peralihan dari tiap proses produksi untuk menghasilkan item yang dipesan. Biasanya meliputi menyiapkan pesanan toko, penjadwalan pekerjaan, menyusun kegiatan pra produksi, ekspedisi, dan penerimaan mutu.

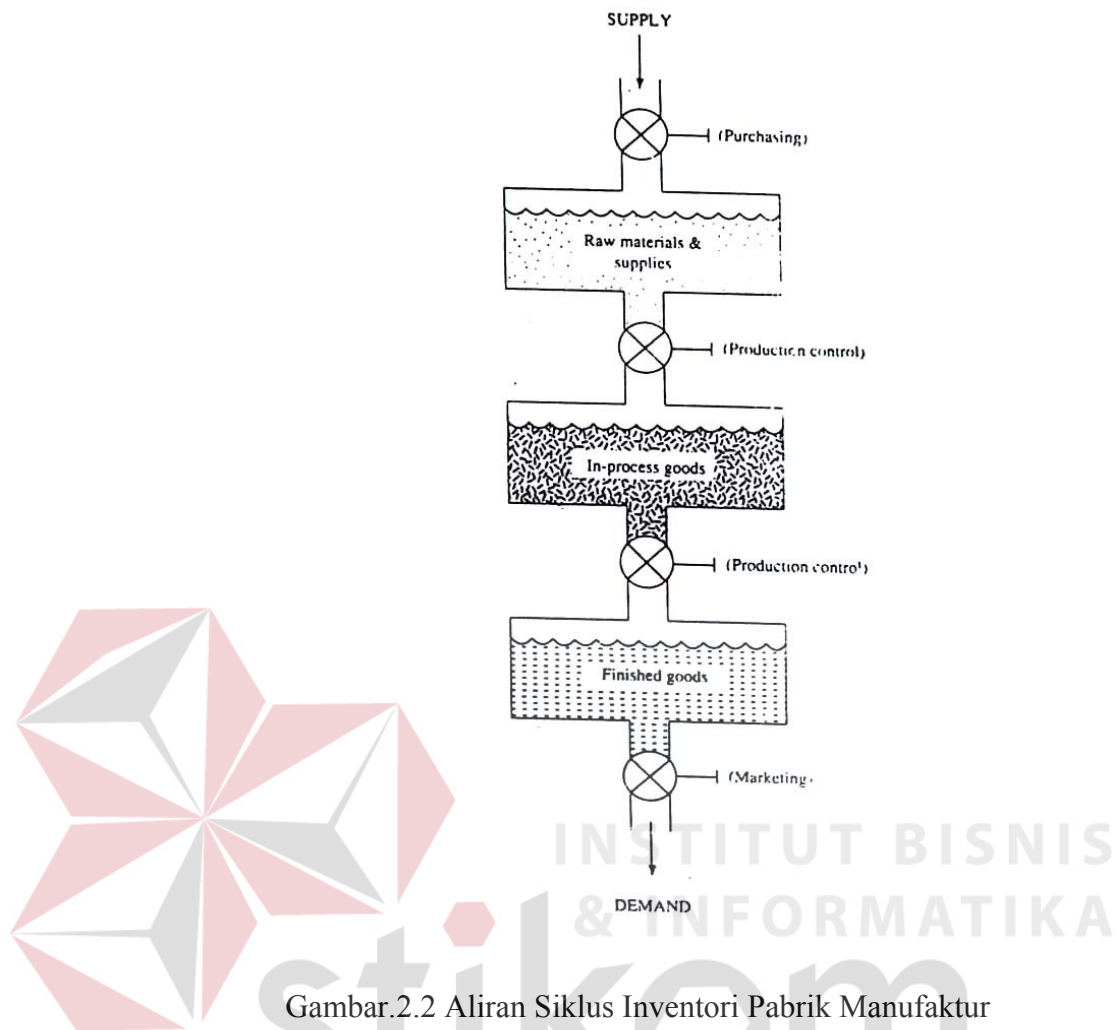
Biaya Penyimpanan (H), memiliki arti yang sama dengan biaya penggudangan, menggolongkan biaya-biaya yang berhubungan dengan penanaman modal dalam inventori dan memelihara investasi fisik di dalam

penyimpanan. Biaya Penyimpanan menyertakan materi seperti biaya-biaya modal, pajak, asuransi, handling, penyimpanan, penyusutan, keusangan, dan pembusukan. Biaya Modal menggambarkan biaya kesempatan atau kemampuan menghasilkan laba yang hilang. Jika dana diinvestasikan ditempat lain, akan diharapkan sebuah kembalian modal investasi.

Biaya Stockout (biaya kehabisan persediaan) adalah konsekuensi ekonomi dari suatu kekurangan eksternal atau internal. Suatu kekurangan eksternal terjadi bila ada pesanan pelanggan yang tidak terpenuhi, kekurangan internal terjadi bila satu permintaan grup atau departemen di dalam organisasi tidak terpenuhi. Kekurangan-kekurangan eksternal dapat menimbulkan biaya-biaya backorder, kerugian laba masa kini (penjualan potensial), dan kerugian laba masa depan (erosi goodwill). Kekurangan-kekurangan internal dapat mengakibatkan kerugian dalam hal produksi (sumber daya menganggur) dan penundaan tanggal penyelesaian. Meningkatnya biaya tergantung pada reaksi pelanggan terhadap kondisi out-of-stock (kehabisan persediaan). Jika permintaan terjadi selama persediaan sebuah item kosong, kerugian ekonomi tergantung pada backorder terhadap kekurangan stok tersebut, dicukupi dengan penggantian item lain, atau dibatalkan. Dalam suatu situasi, penjualan tidak hilang tetapi hanya tertunda beberapa hari dalam pengiriman.

2.1.4 Aliran siklus Inventori

Aliran siklus inventori pabrik manufaktur yang khas digambarkan di dalam gambar 2.2.



Gambar.2.2 Aliran Siklus Inventori Pabrik Manufaktur

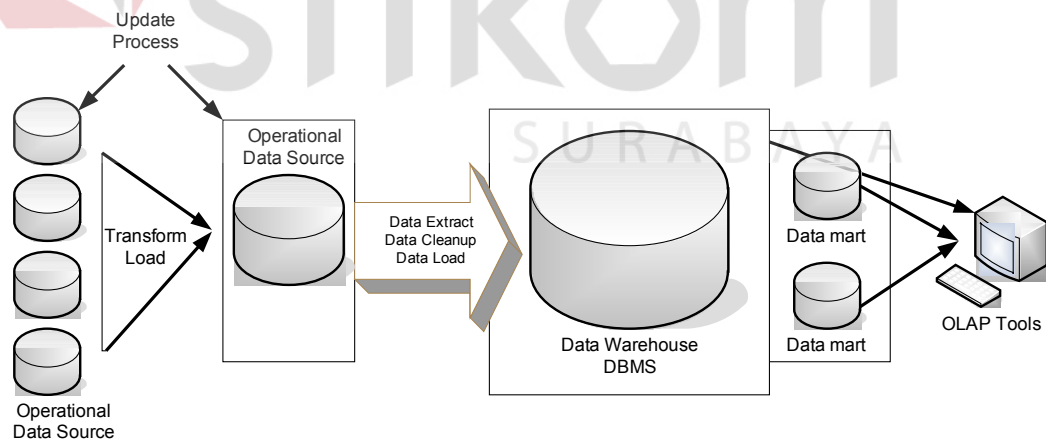
Manajemen inventori melibatkan pengendalian daftar biaya pengiriman barang-barang yang mengalir ke dalam dan ke luar sistem. Pada kenyataannya, ini merupakan suatu masalah penjadwalan atau pemilihan waktu. Pada awal tahap, persediaan dan bahan-bahan diperoleh dari para penyalur. Reservoir dari item-item ini membentuk kolam pertama dari investasi inventori yang harus diatur. Kwantitas dan variasi dari item-item yang dibeli harus diatur waktunya sehingga item-item ini akan menemui permintaan untuk pemanfaatan mereka oleh organisasi. Ketika materi ini dilepaskan sampai ke pabrik, item bergabung dengan reservoir lain dari inventori yang disebut dengan barang dalam proses. Reservoir kedua ini harus diatur dalam hubungan dengan kapasitas fasilitas. Ketika item

meninggalkan kategori barang dalam proses, item-item ini meninggalkan kategori barang-barang yang sedang dikerjakan, mereka masuk reservoir inventori lain yang disebut barang jadi.

Reservoir ketiga ini harus diatur dengan hubungan ke permintaan eksternal. Siklus dapat berlanjut ke sebuah reservoir tambahan jika organisasi melakukan pemeliharaan pada pusat distribusi atau gudang-gudang. Itu menyatakan area-area yang mengatur aliran ke dalam dan ke luar dari kategori-kategori inventori yang berbeda.

2.2 Data Warehouse

Data warehouse adalah penyimpanan data yang single, lengkap, dan konsisten yang didapat dari berbagai macam sumber dan dibuat agar end user dapat dengan mudah memahami serta menggunakannya dalam konteks bisnis (Berson, 1997).



Gambar.2.3 Arsitektur Data Warehouse

Data warehouse didefinisikan sebagai kumpulan subyek data yang terintegrasi, bervariasi, dan *non volatile*. Immon mendefinisikan data warehouse sebagai gabungan dari beberapa sistem yang terintegrasi dan didesain untuk

menyediakan informasi yang dibutuhkan untuk pengambilan keputusan (Berson, 1997).

Fungsi utama dari data warehouse adalah menyediakan data untuk mendukung pembuatan keputusan. Dalam beberapa kasus jenis-jenis dari aplikasi yang telah digunakan misalnya untuk sistem informasi eksekutif (*executive information system*). Data warehouse juga menyediakan berbagai data sebagai input untuk menunjang suatu bisnis dalam melakukan proses analisa bisnis (Haniif, 2008).

Menurut Haniif, ada beberapa karakteristik yang dimiliki oleh sebuah data warehouse, yaitu :

1. Orientasi Subyek.

Suatu data warehouse didesain dan dibangun secara khusus dari database transaksional berdasarkan keperluan perusahaan, semisal data warehouse untuk kastemer, sales, dan lain-lain. Hanya data yang benar-benar diperlukan yang dimasukkan kedalam database.

2. Integrasi data.

Untuk mendapatkan informasi yang diinginkan, data yang ada dalam data warehouse akan dibangun dari beberapa macam sistem yang ada diperusahaan. Kemudian data tersebut ditransformasi dan di*load*. Karena itu ketika disimpan ke dalam data warehouse data tersebut akan diintegrasikan sehingga hanya ada satu cara dan atribut dengan format dan unit yang sama.

3. Nonvolatile.

Pada database transaksional, operasi yang dilakukan adalah operasi update (insert, delete, dan update). Sedangkan dalam data warehouse metode ini

tidak digunakan. Data disimpan ke dalam data warehouse pada periode waktu tertentu setelah dilakukan beberapa perhitungan (*calculation*) dan rangkuman (*summary*).

4. Setiap unit data akan relevan dengan waktu tertentu.

Setiap data yang dimasukkan ke data warehouse pasti memiliki dimensi waktu. Dimensi waktu ini dipergunakan sebagai pembanding dalam perhitungan untuk menghasilkan laporan yang diinginkan. Selain itu, dengan menggunakan dimensi waktu pembuat keputusan dapat mengenal kecenderungan (*trend*) dan pola dari suatu data.

2.2.1 Konsep Multi-dimensional Data

Pada relational database data dikelompokkan dalam sebuah list record. Setiap record mempunyai informasi yang dikelompokkan dalam fields. Pusat dari objek metadata pada multidimensional adalah *cube* atau kubus yang mengandung hubungan struktur dimensi, hirarki, level, dan anggota.

A. Dimensi

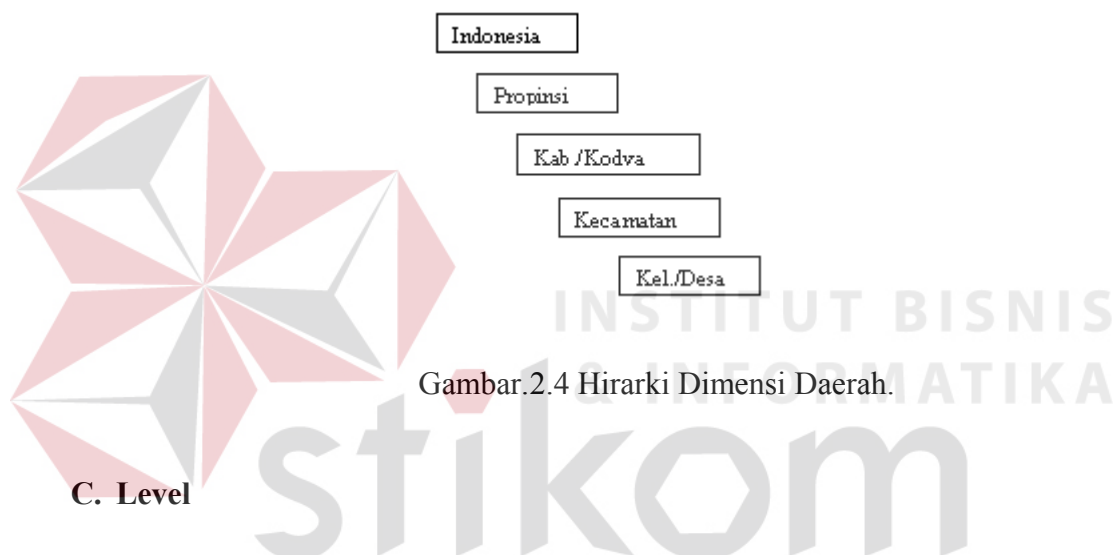
Dimensi merupakan sebuah kategori yang independent dari multidimensional database. Tipe dari dimensi ini mengandung item yang digunakan sebagai kriteria query untuk ukuran database. Contoh pendistribusian obat di suatu daerah. Dimensi **Daerah** = {Jawa Timur, Jawa Barat, DKI Jakarta, Sumatra Selatan, Surabaya, Bandung Jakarta, Palembang, Dago, Caringin, Senen, Matraman}. Dimensi **Waktu** = { Tahun 1999, Tahun 2000, Tahun 2001, Bulan April, Bulan Maret, Bulan Juni, Bulan Juli, Tanggal 1, Tanggal 2, Tanggal 3,

Tanggal 10, Tanggal 12}. Dimensi **Obat** = { Anti Biotik, Vitamin, Ampicilin, Amoxicilin, Enervon C, Redoxon, Hemaviton} (Devi, 2008).

B. Hirarki

Hirarki merupakan bentuk kesatuan dari sebuah dimensi. Sebuah dimensi bisa terbentuk dari multilevel, yang mempunyai parent-child relationship. Hirarki didefinisikan bagaimana hubungan antar level.

Pada Dimensi **Daerah** Mempunyai Hirarki



Gambar.2.4 Hirarki Dimensi Daerah.

C. Level

Level merupakan sebuah kumpulan dalam hirarki. Sebuah dimensi mempunyai multiple layer informasi, setiap layer adalah level. Seperti

- Propinsi = {DKI Jakarta, Jawa barat, Jawa Timur, Sumatra Selatan}
- Kab./Kodya = {Jakarta Pusat, Bandung, Surabaya, Palembang}
- Kecamatan = {Dago, Caringin, Senen, Matraman}

D. Member

Member adalah sebuah item data dalam dimensi, ukuran database dapat didefinisikan menggunakan member yang merupakan parent child seperti :

- {Indonesia} {parent of} {DKI Jakarta, Jawa Barat, Jawa Timur}.

- {DKI Jakarta} {parent of} {Senen, Matraman}.

Dalam sebuah tabel pendistribusian obat.

Tabel 2.1. Tabel Pendistribusian Obat

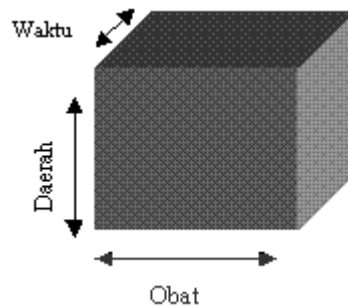
Obat	Daerah	Waktu	Jumlah
Ampicilin	Dago	1990	2359
Ampicilin	Dago	1991	5489
Ampicilin	Senen	1992	2546
Redoxon	Matraman	1990	1254
Redoxon	Senen	1991	623
Redoxon	Dago	1992	2452
Redoxon	Matraman	1992	1254
Enervon-C	Senen	1990	1254
Enervon-C	Dago	1991	1258
Vit.C	Senen	1992	671
Amoxicilin	Dago	1990	7983
Enervon-C	Caringin	1992	568

Data diatas bisa digambarkan dalam bentuk 2 dimensi.

Tabel 2.2. Bentuk tabel 2 dimensi pada penjualan obat.

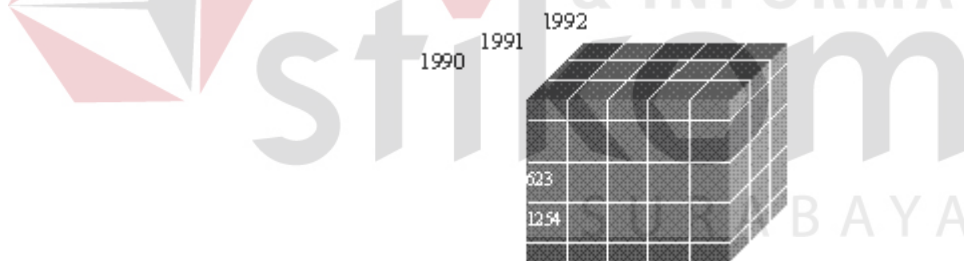
Penjualan Obat			Dimensi 1			
			th	1990	1991	1992
Dimensi 2	Obat	Daerah				
	Ampicilin	Dago		2359	5489	
		Senen				2546
					
	Enervon-C	Dago			1258	
		Senen		1254		
					

Dari data diatas dapat di bentuk suatu analisa data. Data tersebut dapat diambil suatu analisa untuk total penjualan obat merk Ampicilin atau total penjualan obat selama tahun 1990 atau total penjualan untuk daerah matraman. Data tersebut dapat di gambarkan dalam bentuk kubus dimensi.



Gambar 2.5 Bentuk 3 Dimensi

Setiap sel pada kubus merepresentasikan satu nilai variabel jumlah dari vektor-vektor sumbu dari kubus.



Gambar 2.6 Data pada vektor tiap dimensi

Terlihat terdapat hubungan antara dimensi obat dengan dimensi waktu dan dimensi daerah yang menggunakan hirarki pada level kelurahan. Untuk mempermudah pengelompokan data dan pencarian total dari masing-masing informasi yang di inginkan dapat dipilah dalam bentuk hirarki “drill down”? Atau pengelompokan sesuai dengan hirarki pada level yang di inginkan, Obat vs Kabupaten Kotamadya, Obat vs Propinsi.

E. n-dimensional

Untuk bentuk dua dimensi merupakan suatu yang mudah dimengerti. Namun dalam representasi multidimensional bentuk matrik dua dimensi dapat di rubah dalam bentuk tiga dimensi. Pada dua dimensi terdapat dua permukaan (sumbu vektor), tiga dimensi terdapat 6 permukaan (sumbu vektor), pada 4 dimensi terdapat 12 permukaan (sumbu vektor). Maka untuk n dimensi di dapat $n(n-1)$ permukaan (sumbu vektor). Sedangkan sel yang akan terisi seperti contoh pada gambar 3 di atas dimensi Daerah mempunyai 4 item, dimensi waktu punya 3 item, dimensi obat punya 5 item, maka akan terdapat $4 \times 3 \times 5 = 60$ sel yang berkoresponden dengan 60 record relasionalnya. Untuk mendapatkan analisa pada data kubus dapat dilakukan dengan memutar permukaan kubus 90 derajat, maka akan didapat menganalisa dimensi Daerah vs Obat atau dimensi Daerah vs Waktu atau dimensi Waktu vs Obat. (Devi, 2008).

2.2.2 Teknik Pemodelan Data Warehouse

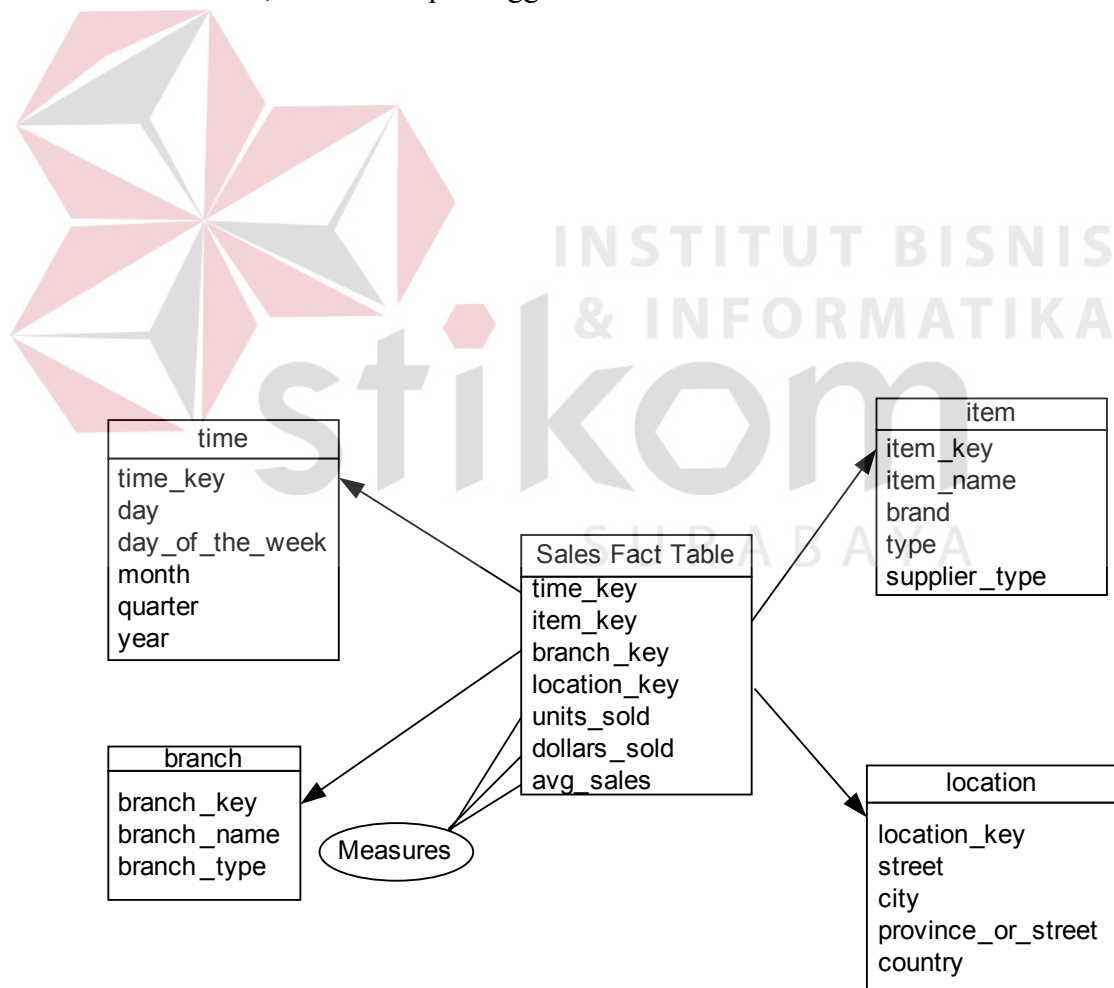
Data warehouse dan OLAP dibangun berdasarkan multidimensional data model. Pada model ini diperlukan tabel fakta dan tabel dimensi. Tabel fakta berisi fakta numerik yang memiliki ciri-ciri : panjang, kurus, dan besar, serta sering berubah dan berguna untuk mengukur (*measure*). Sedangkan tabel dimensi berisi kolom yang bersifat deskriptif, kecil, pendek, dan lebar yang berguna untuk *filtering* (menyaring) dan didasarkan pada atribut dimensi.

Dalam dimensional modeling, ada beberapa pendekatan yang digunakan untuk membuat data warehouse, yaitu:

1. Skema bintang (*star schema*)

Skema ini mengikuti bentuk bintang, dimana terdapat satu tabel fakta (*fact table*) di pusat bintang dengan beberapa tabel dimensi (*dimensional tables*) yang mengelilinginya. Semua tabel dimensi berhubungan dengan ke tabel fakta. Tabel fakta memiliki beberapa key yang merupakan kunci indek individual dalam tabel dimensi.

Skema bintang juga bisa terdiri dari satu atau lebih table fakta. Dikarenakan karena table fakta tersebut ada banyak, misalnya disamping penjualan terdapat table fakta *forecasting* dan *result*. Walaupun terdapat lebih dari satu table fakta, mereka tetap menggunakan table dimensi bersama-sama.

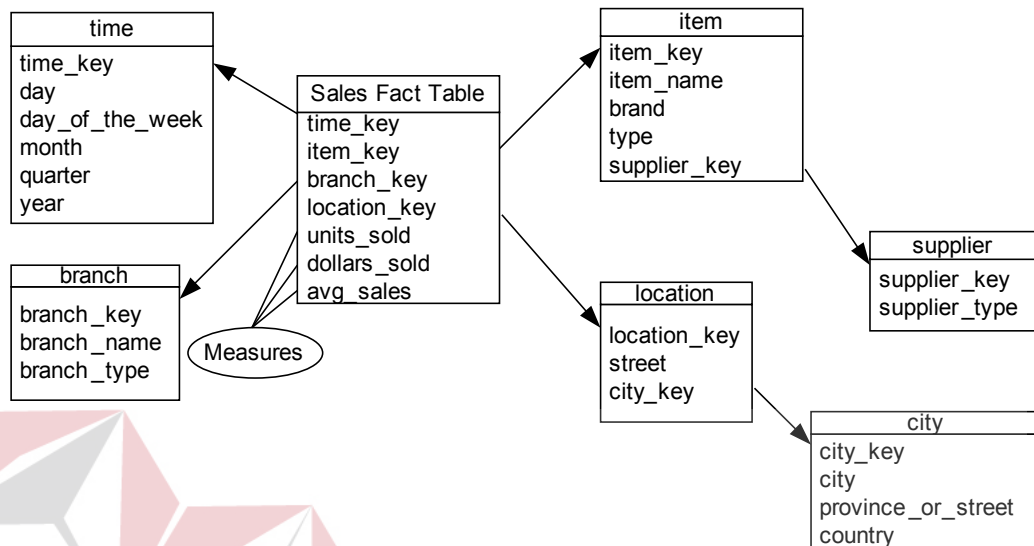


Gambar 2.7 Skema Bintang

2. Skema bola salju (*snowflake Schema*)

Skema bola salju merupakan perluasan dari skema bintang dengan tambahan beberapa tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta.

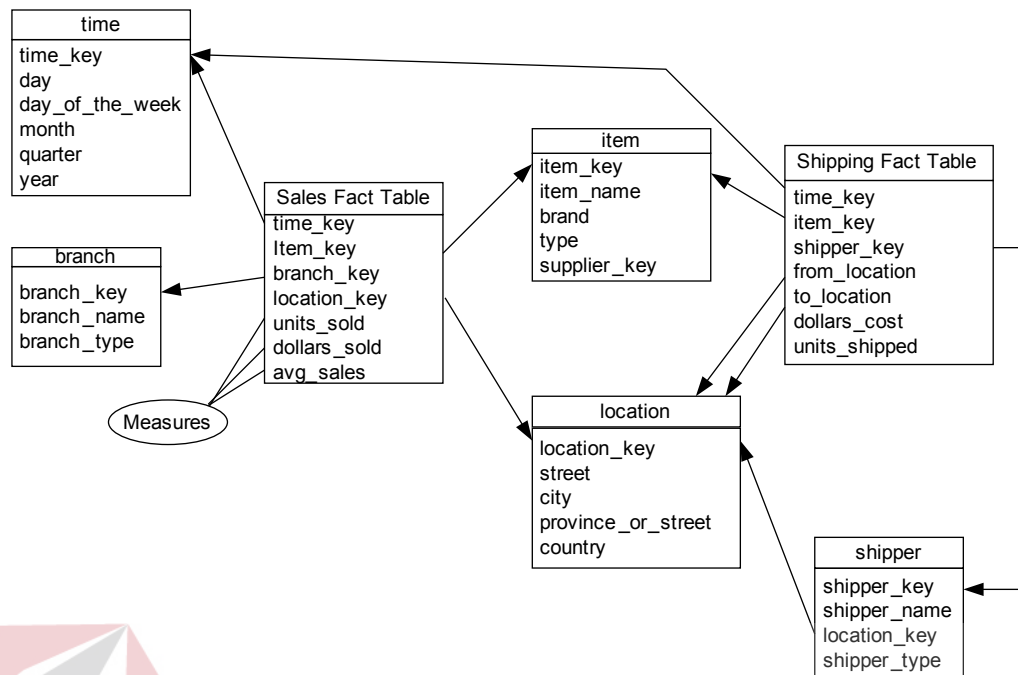
Tabel dimensi tersebut berhubungan dengan tabel dimensi yang lain.



Gambar 2.8 Skema Bola Salju

3. Fact constellations

Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi secara bersama-sama sehingga jika digambarkan akan terlihat seperti sekumpulan bintang. Skema ini juga dikenal dengan *galaxy schema* (Haniif, 2008).



Gambar 2.9 Skema Galaxy

2.2.3 Implementasi Data Warehouse

Beberapa pendekatan yang digunakan untuk mengimplementasikan arsitektur warehouse, yaitu secara *bottom up*, *top down* atau kombinasi antara keduanya.

A. Desain Implementasi Pendekatan *Top Down*

Langkah awal implementasi data warehouse dengan pendekatan *top down* adalah membangun sebuah data warehouse pada semua data perusahaan, setelah itu dilanjutkan dengan membangun data mart yang berisi data warehouse khusus yang merupakan bagian dari data warehouse yang dibangun sebelumnya.

B. Pendekatan *Bottom Up*

Implementasi *bottom up* dimulai dengan membangun data mart untuk menyelesaikan suatu permasalahan tertentu tanpa menunggu dari pengembangan infrastruktur yang telah lengkap. Ketika ada permasalahan yang lain, maka akan

dibuatkan data mart baru. Begitu juga seterusnya. Selanjutnya bermacam-macam data mart tersebut digabungkan menjadi sebuah data warehouse.

C. Pendekatan kombinasi

Dalam beberapa kasus pendekatan tidak harus dengan *bottom up* atau *top down* tetapi juga bisa dilakukan dengan kombinasi keduanya. Hal ini untuk mencari alternatif yang terbaik sesuai dengan kebutuhan (Haniif, 2008).

2.2.4 Extract Transform Load (ETL)

ETL dimulai dari proses extract yaitu mengkonvert data kedalam format yang dapat digunakan untuk proses transformasi. Lalu proses transformasi dilakukan untuk mengaplikasikan rule atau function ke data yang telah diekstrak dari sumber data untuk mendapatkan data yang akan di-load ke target akhir.

A. Extract

Setiap sistem yang terpisah kemungkinan memiliki format yang berbeda. Format yang umum biasanya relasi database dan flat file. Tapi mungkin juga terdapat struktur database non relasional seperti contohnya sumber dari luar misalnya web spider atau onscreen scraping. *Extraction* mengkonvert data kedalam format yang dapat digunakan untuk proses transformasi. Bagian yang paling utama dari ekstraksi adalah parsing dari data yang telah diekstrak, data dianggap benar jika memenuhi pola atau struktur yang diharapkan. Jika tidak kemungkinan data akan ditolak semua.

B. Transform

Tahap yang mengaplikasikan rule atau function ke data yang telah diekstrak dari sumber data untuk mendapatkan data yang akan di-load ke target akhir. Beberapa sumber data kemungkinan akan membutuhkan sangat sedikit

manipulasi data, atau mungkin tidak memerlukan sama sekali. Dalam kasus kasus lain, mungkin akan membutuhkan satu atau lebih dari tipe transformasi, berikut ini digunakan untuk kebutuhan bisnis dan teknikal dari target akhir.

1. Memilih hanya beberapa kolom tertentu untuk load(atau memilih kolom null untuk load).
2. Menterjemahkan nilai nilai kode, misalnya kode P untuk Pria dan W untuk Wanita.
3. Menyandi nilai form seperti 1 untuk Male dan 2 untuk Female
4. Memberikan sebuah hitungan baru, misalnya $sale_amount = quantity * price$.
5. Menyaring data.
6. Mengurutkan data atau penyortiran
7. Untuk join data dari berbagai sumber data contohnya *lookup* dan *merge*
8. Aggregation atau pengumpulan data misalnya untuk menjumlahkan beberapa baris data, total penjualan untuk masing-masing customer, total penjualan untuk masing-masing lokasi.
9. Membangkitkan nilai-nilai kunci.
10. Transposing or Pivoting yaitu mengubah urutan atau berputar (memutar kolom-kolom menjadi baris-baris atau sebaliknya).
11. Splitting, yaitu membagi sebuah kolom menjadi banyak kolom (contohnya memakai tanda koma sebagai pemisah suatu daftar (list) di salah satu kolom, kemudian list yang telah dipisahkan tersebut ditetapkan sebagai string yang kemudian menjadi nilai-nilai yang bersifat individual di kolom-kolom yang berbeda)

12. Menerapkan berbagai macam validasi data, baik validasi yang sederhana maupun yang kompleks. Jika gagal, terjadi penolakan data secara penuh, parsial atau tidak ada penolakan. Dengan demikian, tidak ada data, sebagian data atau semua data diserahkan untuk langkah berikutnya, tergantung pada penanganan exception dan desain rule. Kebanyakan dari transformasi-transformasi yang disebut di atas ada kemungkinan menghasilkan exception/perkecualian, contohnya, bila satu terjemahan kode menguraikan satu kode yang tak dikenal di dalam data yang diekstrak.

C. Load

Fase load memberikan data ke target akhir, biasanya target akhir adalah data warehouse. Tergantung kebutuhan organisasi, proses ini dapat pula memiliki range yang luas. Beberapa data warehouse mungkin dapat setiap minggu menindih data yang ada dengan data komulatif, data yang telah di update, sementara data warehouse lain (atau kemungkinan dapat juga bagian dari datawarehouse yang sama) kemungkinan menambahkan data dari form histori, misalnya perjam. Jangka waktu dan luasnya scope untuk mereplace atau menindih data adalah pilihan design yang bersifat strategik, tergantung waktu yang tersedia dan kebutuhan bisnis itu sendiri.

Sistem-sistem yang lebih kompleks dapat maintenance sebuah histori dan audit trail (jejak2 audit) dari semua perubahan terhadap data yang di load di data warehouse. Mengingat fase load ini berinteraksi dengan database, constraint (batasan) yang telah didefinisikan di skema database maupun trigger yang diaktifkan ketika dilakukan data apply (misalnya unique, referential integrity, mandatory field (field yang harus ada), yang mana juga berperan ke seluruhan

kualitas data performance dalam proses ETL.

2.3 OLAP

OLAP (*On-Line Analytical Processing*) adalah suatu pernyataan yang bertolak belakang atau kontras dengan OLTP (*On-Line Transaction Processing*). OLAP menggambarkan sebuah klas teknologi yang dirancang untuk analisa dan akses data secara khusus. Apabila pada proses transaksi pada umumnya semata-mata adalah pada relational database, OLAP muncul dengan sebuah cara pandang multidimensi data. Cara pandang multimensi ini didukung oleh tehnologi multidimensi database. Cara ini memberikan tehnik dasar untuk kalkulasi dan analisa oleh sebuah aplikasi bisnis.

OLTP mempunyai karakteristik beberapa user dapat *creating, updating, retrieving* untuk setiap record data, lagi pula OLTP sangat optimal untuk *updating* data. OLAP aplikasi digunakan untuk analisa dan mengatur frekuensi level dari agregat/jumlah data. OLAP database biasanya di update pada kumpulan data, jarang sekali dari *multiple source* dan menempatkan kekuatan analisa pada pada back-end pada operasi aplikasi.

Sebab itulah maka OLAP sangat optimal digunakan untuk analisis. *Relational database* merupakan suatu bentuk yang baik untuk mendapatkan suatu record dalam kapasitas jumlah record yang kecil, namun tidak cukup baik dalam mendapatkan suatu record dalam kapasitas jumlah record yang sangat besar serta membuat suatu *summaries* data untuk di analisa, ini memerlukan response time yang lambat dan membutuhkan cukup waktu (Devi, 2008).

Operasi-operasi yang terdapat pada OLAP antara lain :

1. *Slicing* dan *Dicing*

Slicing dan *dicing* adalah operasi untuk melihat data sebagai visualisasi dari kubus. Dengan slicing dan dicing pengguna dapat melihat data dari beberapa perspektif. Pengguna dapat mengekstrak bagian dari data *agregated* dan dapat memeriksa dengan detail berdasarkan dimensi-dimensi yang diinginkan. Data *Agregated* merupakan data praperhitungan (*precalculated*) dalam bentuk rangkuman data (*data summarized*) sehingga query pada kubus (*cube*) lebih cepat. *Slicing* memotong kubus sehingga dapat memfokuskan pada perspektif yang spesifik (pada suatu dimensi). Sedangkan *dicing* memberikan kemampuan untuk melihat pemilihan data pada dua dimensi atau lebih. Yaitu dengan merotasi cube pada perspektif yang lain sehingga pengguna dapat melihat lebih spesifik terhadap data yang dianalisa. Gambar 2.10 sampai dengan 2.15 diambil dari internet <http://www.locationintelligence.net/articles/733.html>

OLAP Cubes						
County	LUNENBURG COUNTY					
Unemployment class	Low					
Household income class	Medium					
Average value of dwelling class	High					
Variables	Population >= 65 years					
Mean	Median	Std. Deviation	Minimum	Maximum	Sum	% of Total Sum
205.00	205.00	14.14	195	215	410	.3%

Gambar 2.10 *Slicing OLAP cubes for Population >= 65 years*

OLAP Cubes						
County	LUNENBURG COUNTY					
Unemployment class	Low					
Household income class	Medium					
Average value of dwelling class	High					
Variables	Total population					
Mean	Median	Std. Deviation	Minimum	Maximum	Sum	% of Total Sum
829.00	829.00	43.84	798	860	1658	.2%

Gambar 2.11 *Slicing OLAP cubes for Total population.*

OLAP Cubes							
County	Total						
Unemployment class	High						
Household income class	Low						
Average value of dwelling class	Low						
Variables	Mean	Median	Std. Deviation	Minimum	Maximum	Sum	% of Total Sum
French mother tongue	13.37	.00	48.96	0	445	1270	3.7%
Aboriginal population	57.05	.00	162.36	0	730	5420	44.7%

Gambar 2.12 *Dicing for two dimensions.*

OLAP Cubes							
County	Total						
Unemployment class	High						
Household income class	Low						
Average value of dwelling class	Low						
Variables	Mean	Median	Std. Deviation	Minimum	Maximum	Sum	% of Total Sum
Total population	518.83	507.00	301.85	48	1200	49289	5.4%
Total dwellings	201.87	188.00	121.35	10	438	19178	5.6%

Gambar 2.13 *Dicing for two dimensions: Total population.*

2. Roll up dan drill down

Drill down dan *roll up* adalah operasi untuk melihat data global atau detail disepanjang level hiraraki dimensi. *Roll up* untuk melihat data secara global atau rangkuman (*summary*). *Drill down* memandu pengguna untuk memperoleh data yang lebih detail. *Drill down* ini biasa digunakan untuk menjawab pertanyaan atas suatu kasus tertentu. Misalnya untuk menjawab pertanyaan ketika sebuah *summary number* (rata-rata atau jumlah) di bawah atau di atas harapan. (Haniif, 2007).

OLAP Cubes							
County	Total						
Unemployment class	Total						
Household income class	Low						
Average value of dwelling class	Total						
Variables	Mean	Median	Std. Deviation	Minimum	Maximum	Sum	% of Total Sum
Total population	249.84	153.00	280.74	0	1200	125669	13.8%
Total dwellings	99.52	53.00	123.23	0	446	50058	14.5%

Gambar 2.14 Rolling-up OLAP cubes.

OLAP Cubes						
County	Total					
Unemployment class	Total					
Household income class	Low					
Average value of dwelling class	Medium					
Variables	Statistics					
	Mean	Median	Std. Deviation	Minimum	Maximum	% of Total Sum
Total population	365.52	233.00	280.54	66	1143	3.8%
Total dwellings	148.27	82.00	121.04	21	413	4.1%
Area	42.9866	3.2091	86.6532	.02	518.67	7.4%
Population >= 65 years	56.21	35.00	52.83	0	235	4.5%
Single-detached house	89.16	70.00	65.94	0	315	3.6%
Households >=6 persons	3.26	.00	5.20	0	15	3.7%
French mother tongue	9.37	.00	47.93	0	465	2.6%
Aboriginal population	4.42	.00	8.90	0	55	3.5%
Unemployment rate	17.372	16.900	11.788	.0	52.6	8.8%
Female unemployment rate	19.694	18.800	19.353	.0	100.0	10.2%
Employment in agriculture	3.63	.00	7.56	0	40	3.6%
Employment in fishing	3.68	.00	9.00	0	50	3.8%
Employment in logging and forestry	3.00	.00	6.08	0	30	5.8%
Bachelor degree or higher	14.68	10.00	14.19	0	70	1.6%
Average household income	9668.72	.00	12603.21	0	28676	1.9%
Average number of rooms per dwelling	6.297	6.400	.985	3.7	8.4	7.2%
Average value of dwelling	69604.01	69597.00	8397.28	56973	85090	6.2%
Count	1.00	1.00	.00	1	1	6.3%

Gambar 2.15 Drilling-down OLAP cubes.

2.4 Oracle Database

Oracle merupakan software database yang banyak dipakai di perusahaan – perusahaan besar di seluruh dunia saat ini. Software ini juga banyak diminati oleh para konsultan pembuat aplikasi yang berkaitan dengan database (Heryanto:1).

Oracle merupakan software database yang bisa menampung serta mengelola data dengan kapasitas yang sangat besar serta perintah SQL nya yang hampir seluruhnya telah memenuhi standar ANSI-92 lebih memudahkan para programmer database dalam membangun aplikasi baik dari sisi '*back end*' maupun dari sisi '*front end*'. Demikian pula bagi seorang administrator yang berkecimpung dalam menangani administrasi database serta bertanggung jawab terhadap keamanan database akan merasa diuntungkan serta dimudahkan dengan software Oracle ini (Heryanto:2:2006).

Oracle dapat diinstal pada versi Windows 2000, Windows XP dan Windows 2003. Proses instalasinya pun cukup mudah dengan persyaratan yang harus dipenuhi minimal memori yang harus tersedia adalah 256 MB dan kapasitas harddisk yang dibutuhkan sekitar 1.3 GB. Sedangkan kapasitas yang diperlukan untuk database awal adalah sekitar 720 MB (Syamsiar:1:2006).

Hal yang harus dipahami dalam struktur database Oracle adalah :

1. ORACLE_BASE. Struktur direktori dasar tempat kita akan melakukan instalasi oracle 10g.
 2. ORACLE_HOME. Direktori yang memuat perangkat lunak Oracle.
 3. ORACLE_SID. Nama *instance* Oracle 10g (*default*-nya adalah ORCL).
- NLS_LANG. Bahasa yang digunakan dan pengaturan – pengaturan karakter yang digunakan (Nugroho:9).

2.4.1 Oracle Warehouse Builder

Oracle Warehouse Builder (OWB) adalah sebuah tool yang komprehensif untuk kegiatan ETL (extract, transform dan load), integrasi secara penuh dari relasional dan dimensional modeling , kualitas data, audit data, dan siklus hidup yang penuh dari manajemen data dan metadata. Mendukung aktivitas manajemen integrasi data dan metadata, termasuk diantaranya:

- Ekstraksi, transformasi, dan loading (ETL) untuk banyak data warehouse.
- Konsolidasi data dari sumber-sumber data yang heterogen.
- Migrasi data dari sistem-sistem sebelumnya.
- Pemodelan data dari struktur relasional dan dimensional.
- Desain dan Pengaturan metadata yang berhubungan dengan perusahaan.
- Pembersihan data untuk memaksimalkan kualitas informasi.
- Pengarsipan data dan audit kualitas.

2.4.2 Oracle Business Intelligence Suite Enterprise Edition

Oracle Business Intelligence Suite Enterprise Edition Plus (Oracle BI EE Plus) adalah sebuah suite yang komprehensif dari produk Bisnis Intelejen enterprise yang memberi area yang penuh dalam kemampuan analisis dan pelaporan (Beacon, 2007).

Dengan menonjolkan arsitektur yang terpadu, berskala tinggi, dan modern, Oracle BI EE Plus menyediakan kecerdasan dan analisa dari sumber-sumber rentang data perusahaan dan memberi kekuatan dalam banyak aplikasi pada komunitas-komunitas terbesar dengan wawasan yang lengkap dan relevan (Beacon, 2007).

Oracle Business Intelligence Suite Enterprise Edition mempunyai keuntungan sebagai berikut :

1. Fungsionalitas BI yang komprehensif dibangun pada infrastruktur yang terpadu. Termasuk di dalamnya dasbor-dasbor yang interaktif, analisa dan query-query khusus yang lengkap, penyampaian dan alert yang proaktif, pelaporan pada skala perusahaan dan pelaporan keuangan, analisa dan penyajian online analytical processing (OLAP), memproduksi pelaporan dengan volume yang tinggi, kecerdasan prediksi yang bersifat real-time, analisis yang tidak bergantung dengan koneksi, juga pengintegrasian dengan Microsoft Office.
2. Pengertian yang mendalam menyebar untuk semua orang, dimana saja.
Menyediakan pengertian mendalam yang relevan untuk semua orang, tidak hanya analis-analis. Semua tingkatan dalam organisasi dapat melihat informasi yang telah dioptimalkan sesuai dengan tugas mereka.
3. Model bisnis yang dipersatukan (unified).
Dibangun di satu model informasi berskala enterprise yang mempersatukan metadata antara tool-tool Oracle BI dan aplikasi-aplikasi analitis untuk TCO paling rendah.
4. Infrastruktur BI yang Hot-Pluggable.
Berintegrasi dengan berbagai tool sumber data, ekstraksi, transformasi, dan load (ETL), aplikasi bisnis utama, server aplikasi, infrastruktur keamanan, teknologi portal, tool analitis dan tool front-end yang lain, serta database yang termasuk IBM DB/2, SAP Business Information Warehouse (BW),

Microsoft Analysis Services; file-file flat; data XML; dan data yang tidak terstruktur.

