BAB 2
TINJAUAN PUSTAKA

2.1. Profil BKKBN Kabupaten Cianjur

Kedudukan BKKBN dalam Keppres No. 38 Tahun 1978 adalah sebagai lembaga pemerintah non-departemen yang berada di bawah dan bertanggung jawab kepada Presiden. Tugas pokoknya adalah mempersiapkan kebijaksanaan umum dan mengkoordinasikan pelaksanaan program KB nasional dan kependudukan yang mendukungnya, baik di tingkat pusat maupun di tingkat daerah serta mengkoordinasikan penyelenggaraan pelaksanaan di lapangan.

Berdasarkan Surat Keputusan Menteri Negara Pemberdayaan Perempuan/ Kepala BKKBN No.74/HK.010/B5/2001 tahun 2001 tentang Tata Kerja BKKBN Provinsi dan kabupaten/Kota, pada tahun 2003 Bupati Cianjur mengeluarkan Surat Keputusan Bupati Cianjur Nomor 18 tahun 2003 tentang Pembentukan Badan Koordinasi Keluarga Berencana Nasional (BKKBN) maka dibentuklah BKKBN Kabupaten Cianjur sebagai Satuan Organisasi Perangkat Daerah.

Pembentukan BKKBN Kabupaten Cianjur ini bertujuan untuk lebih mensejahterakan Keluarga yang ada di wilayah Kabupaten Cianjur, melalui :

1. Menggerakkan dan Memberdayakan Seluruh Masyarakat dalam Program KB.
2. Meningkatkan Ketahanan dan Kesejahteraan Keluarga melalui Pelayanan KB.
3. Memperkuat SDM Operasional Program KB.

2.1.1. Visi dan Misi BKKBN Kabupaten Cianjur

Visi dan misi dari Badan Kependudukan dan Keluarga Berencan Kabupaten Cianjur adalah sebagai berikut:

1. Visi

   Penduduk Kabupaten Cianjur Tumbuh Seimbang Tahun 2015.

2. Misi

   Mewujudkan Pembangunan Berwawasan Kependudukan dan Mewujudkan Keluarga Kecil Bahagia Sejahtera di Kabupaten Cianjur.

2.1.2. Logo BKKBN Kabupaten Cianjur



Gambar 2.1 Logo BKKBN Kabupaten Cianjur

a. Filosofi Logo

Logo baru BKKBN adalah hasil dari adaptasi logo terdahulu dengan tujuan untuk menegaskan eksistensi dan peran BKKBN sebagai lembaga yang mempunyai peranan penting dalam masalah kependudukan di Indonesia. Logo BKKBN merupakan paduan dari elemen-elemen yang tidak dapat dipisahkan yang merupakan perwakilan dari seluruh kinerja BKKBN sebagai lembaga yang menjaga keseimbangan Kualitas, Kuantitas dan Mobilitas penduduk Indonesia dalam mewujudkan kesempurnaan yang disimbolkan dalam satu kesatuan logo. Ikon yang terdiri dari bapak, ibu, dan dua orang anak yang berada di bawah naungan lengkungan warna biru muda merupakan simbol terdepan dari satu keluarga indonesia yang dinamis dan berkualitas, pada logo terdapat lengkungan berwarna biru yang merupakan cakrawala biru yang luas cerminan kinerja BKKBN yang luas dalam menjaga keseimbangan keluarga Indonesia. Bentuk dan susunan tipografi BKKBN digambarkan dengan formasi yang solid dengan warna biru muda pada huruf B, N dan huruf k, k, b berwarna biru tua yang merupakan satu kesatuan yang harmonis untuk mewujudkan visi dan misi BKKBN yang selaras dan satu tujuan secara harmonis demi menuju penduduk tumbuh seimbang.

b. Warna logo

Logo BKKBN terdiri dari 2 warna yaitu biru muda dan biru tua. Notasi warnanya adalah:

Warna Pure Blue, Pantone Process Cyan C, C100 M0 Y0 K0

Warna ini memiliki arti lembaga tulus melakukan tugasnya.

Warna Elegant Blue, Pantone 2955C, C100 M50 Y0 K40

Warna ini memiliki arti lembaga terpercaya, berwibawa dan satu kesatuan utuh.

2.1.3. Tugas Pokok dan Fungsi

1. Tugas

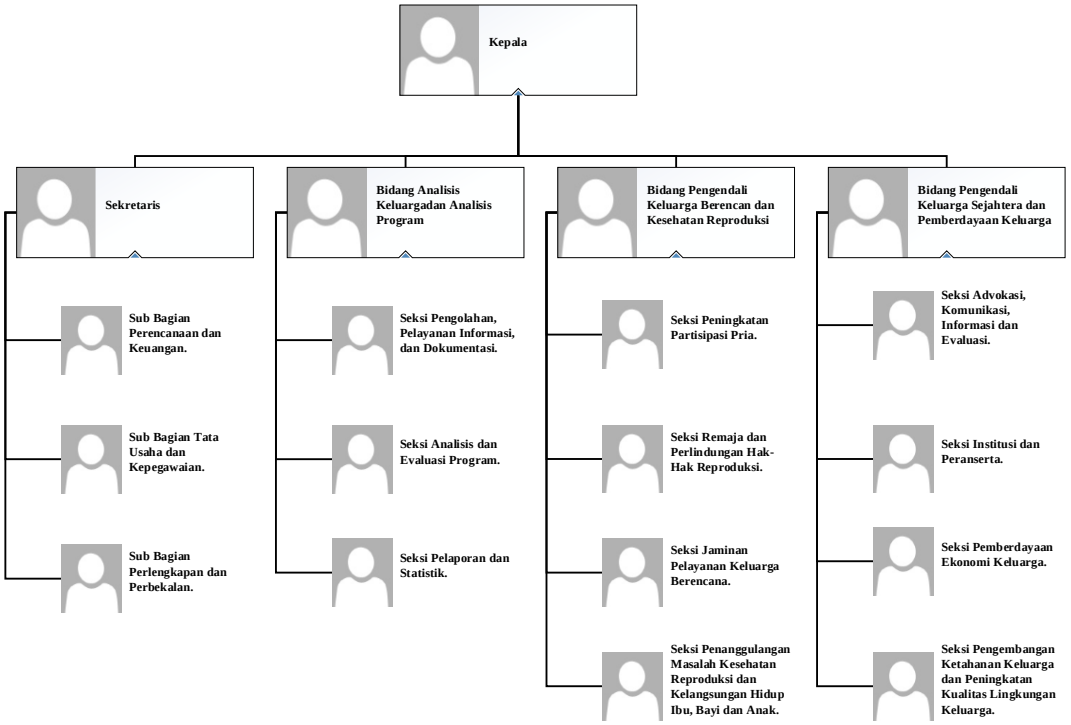
a. Melaksanakan tugas pemerintahan dibidang keluarga berencana dan keluarga sejahtera sesuai dengan ketentuan peraturan perundang-undangan yang berlaku

2. Fungsi

a. Pengkajian dan penyusunan kebijakan nasional di bidang Keluarga Berencana dan Keluarga Sejahtera.
b. Koordinasi kegiatan fungsional dalam pelaksanaan tugas BKKBN.
c. Fasilitasi dan pembinaan terhadap kegiatan instansi pemerintah, swasta, LSOM dan masyarakat dibidang Keluarga Berencana dan Keluarga Sejahtera.
d. Penyelenggaraan pembinaan dan pelayanan administrasi umum dibidang perencanaan umum, ketatausahaan, organisasi dan tatalaksana, kepegawaian, keuangan, kearsipan, hukum, persandian, perlengkapan dan rumah tangga.

2.1.4. Struktur Organisasi

Struktur organisasi adalah pola hubungan antara bagian-bagian dari instansi atau menggambarkan dengan jelas pemisah kegiatan pekerjaan antar bagian dalam suatu instansi. Berikut struktur organisasi BKKBN Kabupaten Cianjur dapat dilihat pada **Gambar 2.2**.



Gambar 2.2 Struktur Organisasi BKKBN Kabupaten Cianjur

2.1.4.1. Kelompok Jabatan Fungsional

Jumlah Pegawai BKKBN Kabupaten Cianjur :

1. DT II / Tingkat Kabupaten : 45 Orang
2. Petugas Fungsional dan PLKB/Lapangan : 280 Orang

Jumlah Seluruh Personil : **225 Orang**

Terdiri dari :

1. DT II / Tingkat Kabupaten :

   a. Kepala Kantor : 1 Orang
   b. Sekretaris : 1 Orang
   c. Kepala Bidang : 3 Orang
   d. Kepala Sub Bagian : 2 Orang
   e. Kepala Seksi : 10 Orang
   f. Staf : 25 Orang

g. Pengemudi : 2 Orang
h. Pesuruh : 1 Orang
Jumlah : **45 Orang**

2. Petugas Lapangan KB :

a. Pengendali Program tingkat Kecamatan : 1 Orang
b. Staf P.PLKB tingkat Kecamatan : 1 Orang
c. Penyuluh KB Tingkat Desa/Keluarahan : 3 Orang
d. PLKB Tingkat Desa/Kelurahan : 2 Orang
Jumlah : **180 Orang**

2.2. Landasan Teori

2.2.1. Data Warehouse

Data warehouse adalah sebuah tempat penyimpanan data yang lengkap dan konsisten dimana data yang disimpan itu saling berelasi dan dirancang berdasarkan query dan analisis daripada proses transaksinya [2]. Data tersebut ditransformasikan menjadi sebuah informasi yang dapat diakses kapan saja dan selalu *up to date*. Informasi ini kemudian akan digunakan untuk dianalisis untuk menghasilkan informasi baru.

Data warehouse juga bisa dikatakan sebagai kumpulan data yang berorientasi subjek, terintegrasi, tidak dapat di *update*, memiliki dimensi waktu yang digunakan untuk mendukung proses manajemen pengambilan keputusan dan kecerdasan bisnis. Berdasarkan definisi tersebut, maka data *warehouse* memiliki karakteristik [2] sebagai berikut :

1. *Subject oriented* (berorientasi subjek)

   Data *warehouse* dirancang untuk membantu *user* dalam pengambilan keputusan. Contohnya untuk mengetahui tentang data penjualan perusahaan. Kita bisa membangun *data warehouse* yang berfokus pada penjualan. Dengan menggunakan *data warehouse*, kita bisa menjawab

pertanyaan seperti “Siapa pembeli terbaik untuk barang ini tahun lalu?”. Kemampuan untuk mendefinisikan sebuah data *warehouse* sebagai sebuah subjek, dalam hal ini penjualan, membuat data *warehouse subject oriented*.

2. *Integrated* (terintegrasi)

   Data warehouse dikonstruksikan dengan cara mengintegrasikan sejumlah sumber data yang berbeda. Data yang terintegrasi menyebabkan data tersebut lebih konsisten, sehingga lebih mudah dipahami oleh para pembuat keputusan.

3. *Time-variant*

   Data warehouse harus bisa menghasilkan informasi dari sudut pandang historical (misalnya informasi 5-10 tahun yang lalu atau bahkan lebih). Atau bisa dikatakan bahwa data *warehouse* berfokus pada perubahan setiap waktunya.

4. *Non-volatile*

   Data yang ada dalam *data warehouse* tidak bisa di-*edit* ataupun di-*update*.

Data warehouse dibuat untuk melayani *user* (*analyst* dan pengambil keputusan). Sehingga *data warehouse* wajib dirancang sesuai dengan persyaratan [1] berikut :

1. Harus bisa memberikan kepuasan kepada setiap *user*.
2. Memiliki *function* sendiri tanpa mengganggu OLTP *systems*.
3. Menyediakan pusat tempat penyimpanan data yang konsisten.
4. Menjawab setiap *complex queries* dengan cepat.
5. Menyediakan berbagai analisis *tools* yang kuat, seperti OLAP dan data *mining*.

Sebagian besar *data warehouse* yang sukses selain memenuhi persyaratan di atas juga memiliki beberapa karakteristik [1] seperti :

1. Berdasarkan model dimensional.
2. Mengandung *historical* data.
3. Terdiri dari *detailed* dan *summarized* data.

4. Tetap mempertahankan konsistensi data walaupun berasal dari sumber yang berbeda.
5. Fokus dalam *single subject*, seperti penjualan, keuangan, atau inventarisasi.

Data warehouse bisa dikatakan sebagai suatu salinan dari OLTP (*On-Line Transaction Processing*) yang terstruktur yang digunakan untuk memenuhi kebutuhan analisis, *reporting*, maupun data *mining* [1]. OLTP sendiri adalah sebuah proses yang menitikberatkan pada transaksinya, seperti input data dan lain-lain. Pada **Tabel 2.1** akan memperlihatkan perbedaan OLTP *systems* dan data *warehouse*.

Tabel 2.1 Perbedaan OLTP dan Data Warehouse [3]

OLTP	**Data *Warehouse***
Dirancang untuk operasi *real-time* bisnis	Dirancang untuk analisis dari suatu bisnis berdasarkan atribut dan kategori
Menangani data saat ini	Menangani data saat ini dan data masa lalu
Data disimpan pada beberapa *platform*	Data disimpan pada satu platform saja
Data diorganisir berdasarkan fungsi atau operasinya	Data diorganisir berdasarkan subjek
Prosesnya bersifat berulang (*loop*)	Prosesnya dilakukan setiap saat dan harus berorientasikan waktu (*historical*)
Untuk operasional	Untuk managerial
Berorientasi pada transaksi	Berorientasi pada analisis

2.2.1.1. Tujuan Data Warehouse

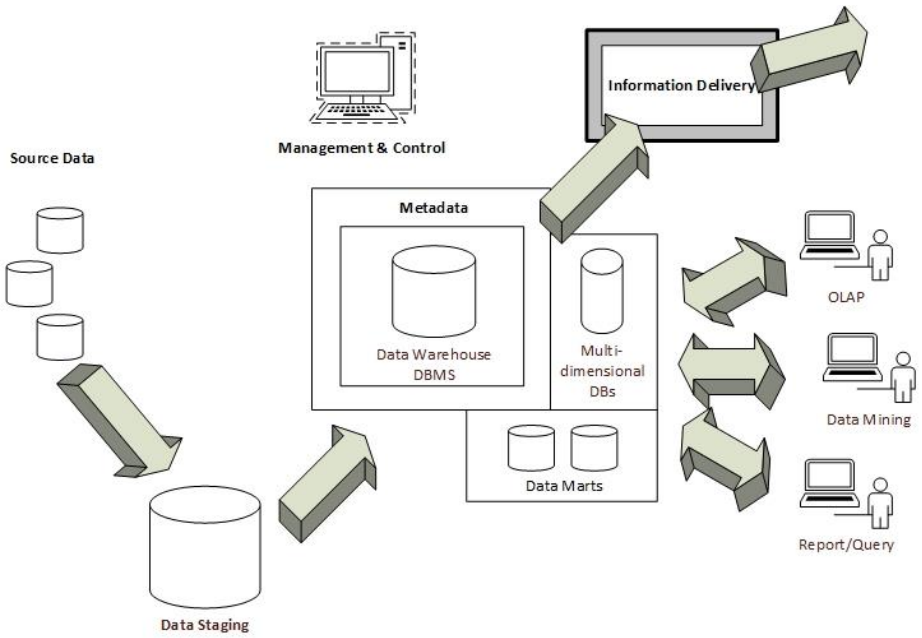
Tujuan dari *data warehouse* adalah sebagai berikut [4] :

1. Memberikan kemudahan untuk mengakses informasi yang ada.
2. Kemudahan disini berbicara tentang efisiensi. *Data warehouse* harus efisien sehingga dengan mudah dipahami oleh *user* bukan hanya oleh *devoleper* saja. Selain itu, pengguna juga dapat mengkombinasikan data dalam *data warehouse* dengan berbagai cara (*slicing and dicing*). Untuk mengakses *data warehouse* disarankan sebaiknya dapat dilakukan dengan sederhana dan mudah dioperasikan.

3. Menyediakan informasi yang konsisten.
4. *Data warehouse* hanya berisi informasi-informasi yang relevan bagi kebutuhan *user* untuk pengambil keputusan. Oleh karena itu, kredibilitas data yang terdapat dalam *data warehouse* harus dapat dipertanggungjawabkan.
5. Mampu beradaptasi dan tahan terhadap perubahan.
6. Perubahan-perubahan yang terjadi harus dapat diatasi oleh *data warehouse*. Dengan kata lain, *data warehouse* harus dirancang agar mampu menghadapi setiap perubahan dengan terencana. Hal ini berarti perubahan yang terjadi tidak boleh merusak atau mengganggu data dan aplikasi yang telah ada sebelumnya.
7. Mampu mengamankan informasi.
8. Informasi yang tersimpan dalam *data warehouse* harus tersimpan dengan aman. Dengan kata lain, informasi tersebut tidak boleh sampai jatuh ke tangan yang salah. Oleh karena itu, *data warehouse* harus mampu mengendalikan setiap akses dari informasi yang ada.
9. Mampu memberikan dukungan dalam pengambilan keputusan.
10. Ini merupakan tujuan yang paling penting dan harus ada dalam setiap pembuatan *data warehouse*. Data warehouse bisa digambarkan sebagai kumpulan teknologi pendukung keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan informasi, dapat membuat keputusan dengan cepat dan tepat.
11. *User friendly*.
12. Seperti pada tujuan *data warehouse* pertama, *data warehouse* harus dirancang agar dapat dioperasikan dengan mudah oleh *user*. Tidak seperti sistem operasional dimana seringkali *user* tidak memiliki pilihan yang lain kecuali menggunakan sistem baru, akan tetapi *user data warehouse* biasanya merupakan pilihan. Oleh karena itu, proses penentuan *user data warehouse* merupakan faktor yang sangat penting.

2.2.1.2. Arsitektur Data Warehouse

Arsitektur pada data *warehouse* dikelompokkan menjadi 5 bagian seperti **Gambar 2.3** [5] :



Gambar 2.3 Arsitektur *Data Warehouse* [5]

Penjelasan dari **Gambar 2.3** [5] adalah sebagai berikut:

1. Source Data

   Merupakan sumber data, atau bisa dibilang bahwa darimana data itu berasal. Untuk membangun *data warehouse* BKKBN Kabupaten Cianjur maka source data berasal dari operasional *system* atau OLTP *database*.

2. Data Staging

   Merupakan sebuah proses yang diperlukan sebelum *data source* masuk ke dalam *data warehouse*. Proses ini dinamakan proses ETL (*Extracting*, *Transformation*, *Loading*).

3. Data Warehouse

   Merupakan tempat penyimpanan data yang multidimensi, dimana data yg tersimpan berupa *metadata*, *summary data* dan *raw data*.

4. Data Marts

Merupakan bagian dari *data warehouse*. Seluruh *data mart* jika digabungkan akan menjadi satu *data warehouse*.

5. Users

   Merupakan pengguna yang akan menggunakan *data warehouse*. *Users* bisa memiliki tugas yang berbeda-beda, ada yang untuk analisis, ada yg untuk *reporting*, dan ada yang untuk membentuk *data mining*. Intinya informasi yang berasal dari *data warehouse* akan diolah menjadi "sesuatu" menggunakan *tools* yang ada yang bisa berguna untuk pengambilan keputusan.

2.2.1.3. Dimensional Data Warehouse

Kebutuhan *user* dan realitas data yang menjadi faktor penentu untuk merancang *dimensional* model *data warehouse*, seperti bisnis apa yang paling diperlukan, detailnya seperti apa dan dimensi-dimensi serta fakta-fakta apa yang harus diikutkan [2].

Maka *dimensional* model harus disesuai dengan kebutuhan dari *user*. Model juga harus dirancang sedemikian rupa agar dapat bertahan dan dapat beradaptasi dari segala perubahan yang akan terjadi. Desain modelnya yang dihasilkan dibentuk menjadi *database* relasional yang mendukung OLAP *cubes* untuk menyediakan secara "*instant*" hasil *query* untuk analis.

1. Tabel Dimensi (Dimension Tables)

   Tabel dimensi menjelaskan tentang entitas bisnis dari suatu *enterprise* [2]. Tabel dimensi umumnya berisi data keterangan, dimana data tersebut jarang sekali mengalami perubahan.

2. Tabel Fakta (Fact Tables)

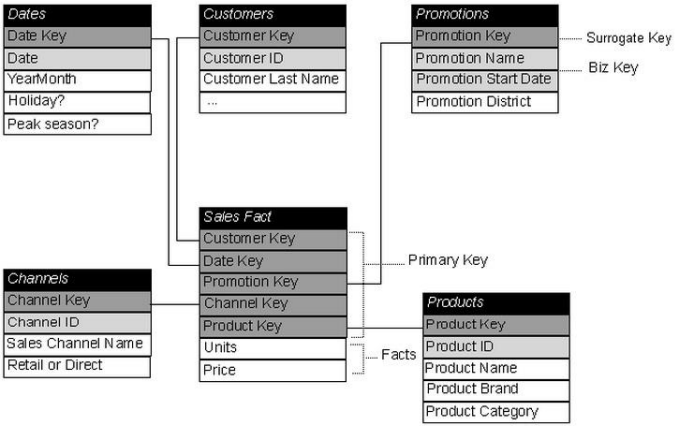
   Tabel fakta merupakan sebuah tabel yang menjelaskan tentang transaksi bisnis dari suatu *enterprise* biasanya disebut tabel *detail* [2]. Tabel fakta umumnya berisi data yang berkaitan langsung dengan proses bisnisnya.

3. Skema Dimensional Model

   Berikut ini adalah beberapa skema yang biasa digunakan untuk merancang suatu *data warehouse* :

a) Skema star

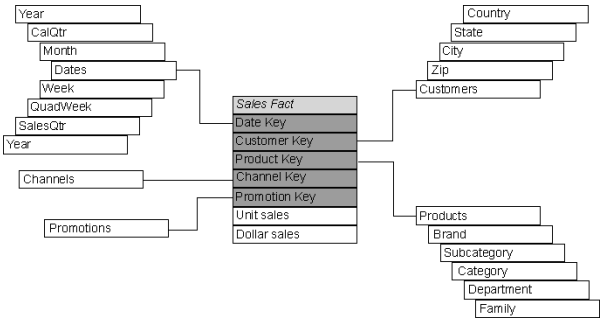
Suatu skema disebut skema *star* jika seluruh tabel dimensi dihubungkan secara langsung ke tabel fakta dan satu tabel fakta wajib memiliki relasi minimal dengan satu tabel dimensi [2]. **Gambar 2.4** adalah contoh gambar dari skema *star*.



Gambar 2.4 Skema *Star* [3]

b) Skema snowflake

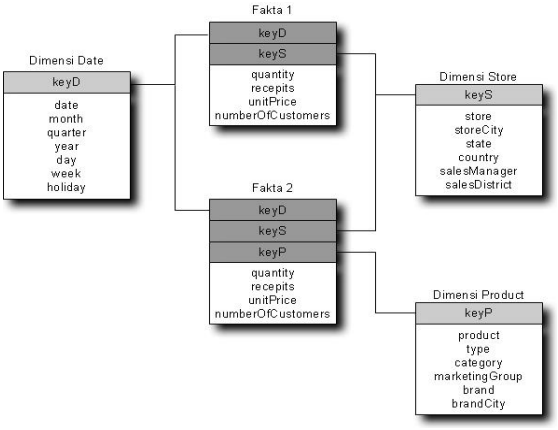
Suatu skema disebut skema *snowflake* jika satu atau lebih tabel dimensi tidak berhubungan langsung dengan tabel fakta tetapi harus berhubungan melalui tabel dimensi lain [2]. **Gambar 2.5** adalah contoh gambar dari skema *snowflake*.



Gambar 2.5 *Snowflake* [3]

c) Skema constellation

Suatu skema dikatakan sebagai skema *constellation* jika ada satu atau beberapa tabel dimensi yang dipakai bersamaan oleh satu atau lebih tabel fakta [3]. **Gambar 2.6** adalah contoh gambar dari skema *constellation*.

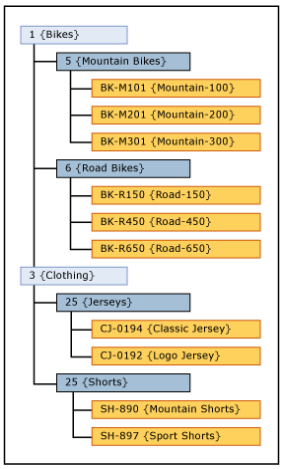


Gambar 2.6 *Constellation* [6]

2.2.1.4. Hirarki (Hierarchies)

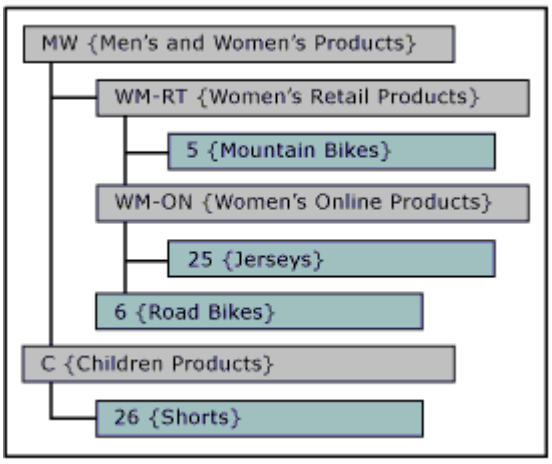
Hirarki merepresentasikan *relationships* antar tiap entitas ataupun tiap anggota entitasnya. Hirarki bisa juga dikatakan sebagai *parent-child relationships* karena hirarki memandang sesuatu melalui berbagai sudut pandang. Hirarki dibagi menjadi:

1. *Derived Hierarchies*, merupakan suatu hirarki yang memiliki *relationships* antar entitas, sebagai contoh entitas kategori berhubungan dengan entitas produk, maka ada suatu cara untuk membangun *derived relationships* berdasarkan nilai dari atributnya. Contoh lainnya bisa dilihat pada **Gambar 2.7**.



Gambar 2.7 *Derived Hierarchies*

2. *Explicit Hierarchies*, merupakan suatu hirarki yang merepresentasikan *relationships* antar tiap member dalam sebuah entitas. Biar lebih jelas untuk *explicit hierarchies* bisa dilihat pada **Gambar 2.8**.



Gambar 2.8 *Explicit Hierarchies*

2.2.1.5. Proses ETL dalam Data Warehouse

Proses ETL atau biasa disebut *Extract*, *Transform*, dan *Load* merupakan proses pengubahan data dari OLTP *database* menjadi *data warehouse*. Jika dilihat dari arstitektur *data warehouse*, proses ETL ini merupakan proses yang berada di *data staging*.

Proses ETL merupakan proses untuk mengubah, memformat ulang serta mengintegrasikan data yang berasal dari satu atau beberapa OLTP *systems* [6].

1. Extraction

   Extraction merupakan sebuah proses dimana proses tersebut mencari sumber data kemudian dengan menggunakan beberapa kriteria yang sudah diberikan untuk memilah data dan juga untuk mencari data yang berkualitas, kemudian data tersebut diangkut ke file lain atau database [6].

2. Transformation

   Data transformation merupakan suatu fase yang terjadi ketika data sudah menjadi raw data (hasil extraction) diubah menjadi bentuk yang sudah ditetapkan dimana bentuk tersebut harus bisa digunakan dalam

data warehouse [5]. Berikut ini adalah beberapa proses dasar yang harus ada dalam data *transformation* :

a) Selection

Memilih atau memilah data hasil dari *extraction*.

b) Splitting/Joining

Splitting/*joining* meliputi tipe-tipe manipulasi data yang perlu dilakukan pada proses *selection*.

c) Conversion

Proses ini merupakan tahapan paling penting. Pada tahap *conversion*, data hasil *selection* kemudian akan diubah menjadi data yang layak digunakan pada *data warehouse*.

d) Summarization

Tahap ini merupakan tahap pembentukan model yang akan ditampilkan kepada user.

e) Enrichment

Tahap ini merupakan tahap pembentukan kembali serta penyederhanaan *field* yang ada untuk membuat *field* tersebut lebih berguna pada *data warehouse*.

3. Loading

Loading adalah suatu proses pemindahan data secara fisik dari OLTP *systems* ke dalam data *destination* atau *data warehouse*. Operasi *loading* terdiri dari memasukkan *record* ke dalam bermacam-macam dimensi dan *fact* tabel yang ada pada data destination atau *data warehouse* [4].

2.2.1.6. Data Warehouse Tools

Berikut ini adalah *tools* yang digunakan *user* setelah *data warehouse* terbentuk dengan tujuan yang berbeda-beda [1]:

1. OLAP (On-Line Analytical Processing)

OLAP merupakan salah satu *data warehouse* tools untuk melakukan analisis data. OLAP sendiri adalah suatu teknologi yang dirancang

untuk memberikan kinerja yang unggul untuk *ad hoc business intelligence queries* [1]. OLAP dirancang untuk beroperasi secara efisien dengan data yang terorganisir sesuai dengan model dimensi umum yang biasa digunakan dalam *data warehouse*.

Tidak ada *data warehouse modern* yang selesai dengan sempurna tanpa fungsionalitas OLAP. Tanpa OLAP, kita tidak dapat memberikan users seluruh kemampuan untuk melakukan analisis multidimensional, untuk melihat informasi dari segala sudut pandang, dan untuk membuat keputusan yang bersifat kritikal. Oleh karena itu, OLAP sangat krusial [5]. Berikut ini adalah karakteristik paling mendasar dalam OLAP *systems* [5] :

a) Memberikan para pelaku bisnis pandangan logis yang multidimensi dari data yang ada dalam *data warehouse*.
b) Memfasilitasi *query* interaktif dan kompleks analisis kepada *users*.
c) Memungkinkan *user* untuk *drill-down* atau *roll-up* data yang ada baik untuk *single dimension* maupun untuk *multi dimension*.
d) Memberikan kemampuan untuk melakukan perhitungan dan perbandingan yang rumit.
e) Hasilnya bisa dipresentasikan dalam bentuk yang lebih berarti, seperti grafik atau tabel.

Kegunaan OLAP [5] antara lain:

a) Meningkatkan produktivitas dari bisnis manajer, eksekutif dan analis.
b) Memanfaatkan OLAP dengan baik bisa membuat *users* bisa dengan percaya diri membuat analisis mereka sendiri tanpa bantuan IT *assistance*.
c) Keuntungan untuk IT *developers* yaitu penggunaan OLAP bisa sangat membantu mempercepat kinerja dari aplikasinya sendiri.
d) Meningkatkan efisiensi kerja.

OLAP dapat digunakan untuk melakukan hal-hal seperti [4]:

a) Consolidation (roll-up)

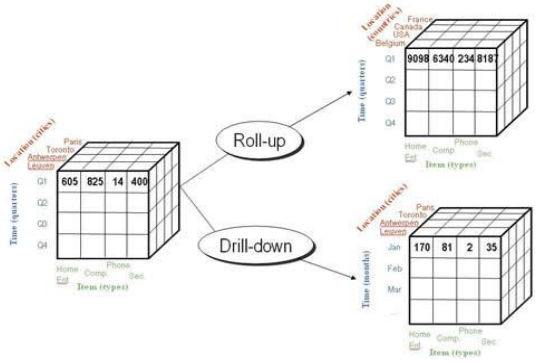
Konsolidasi melibatkan pengelompokkan data untuk melihat data secara global atau rangkuman (*summary*).

b) Drill-down

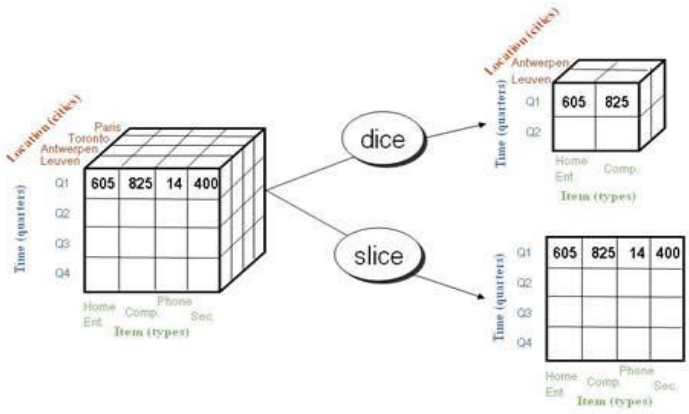
Suatu bentuk yang merupakan kebalikan dari konsolidasi, untuk mendapatkan lebih detail tentang suatu dimensi serta bisa dikatakan sebagai suatu navigasi dari tingkat yang lebih umum ke tingkat yang lebih spesifik. Untuk lebih jelasnya *roll-up* dan *drill-down* dapat dilihat pada **Gambar 2.9**.

c) Slicing and 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 *agregrated* dan dapat memeriksa dengan detail berdasarkan dimensi-dimensi yang diinginkan. *Data Agregrated* 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 dianalisis. Untuk lebih jelasnya slicing and dicing bisa dilihat pada **Gambar 2.10**.



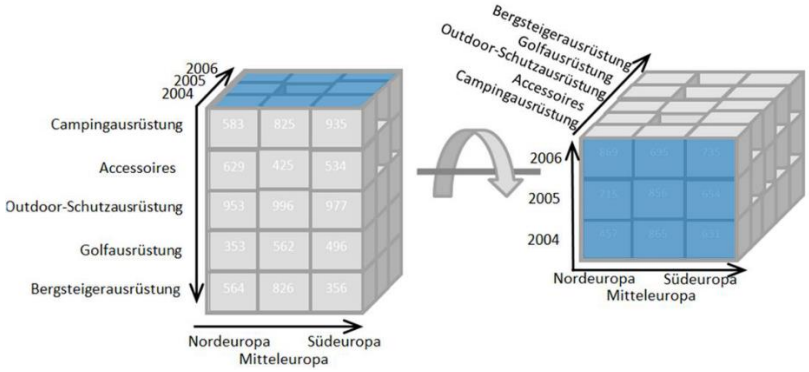
Gambar 2.9 *Roll-up* dan *Drill-down*



Gambar 2.10 *Slicing* and *Dicing*

d) Pivot

Menampilkan nilai-nilai ukuran dalam tata letak tabel yang berbeda dan juga bisa mengatur kembali dimensi dalam OLAP *cube*. Untuk lebih jelasnya *pivot* bisa dilihat pada **Gambar 2.11**.



Gambar 2.11 *Pivot*

Untuk memodelkan OLAP kepada *users*, kita gunakan OLAP *cube*. OLAP *cube* adalah bagian utama dari OLAP yang berisi kumpulan data yang banyak dan disatukan (*agregasi*) untuk mempercepat hasil *query* [4]. OLAP *cube* contohnya seperti pada **Gambar 2.9**, **Gambar 2.10**, **Gambar 2.11**.

2. Reporting

Reporting tools merupakan *tools* yang digunakan untuk mempermudah user memperoleh data yang sudah lama ataupun data sekarang dan melakukan beberapa standard analisis statistik [5]. Data yang dihasilkan

dari *reporting tools* bisa berupa bentuk laporan biasa dan juga bisa berupa grafik.

3. Data mining

   Data mining merupakan teknologi yang mengaplikasikan algoritma yang canggih dan kompleks untuk menganalisis data dan mencari informasi yang menarik dari kumpulan data tersebut. Perbedaan mendasar antara OLAP dan *data mining* yaitu terletak pada apa yang akan dianlisisnya. Pada OLAP, yang dianalisis adalah modelnya, tetapi pada *data mining* yang dianalisis adalah datanya (harus berjumlah besar) [1].

2.2.2. Microsoft SQL Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk *Microsoft*. Bahasa *query* utamanya adalah *Transact-SQL* yang merupakan implementasi dari *SQL* standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya *SQL Server* digunakan di sunia bisnis yang memiliki basis data yang berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar [7]. Pengguna yang menggunakan *SQL Server* dapat menyimpan banyak data dan dapat untuk diimplementasi untuk kepentingan bisnis dan perusahaan.

2.2.3. SSIS (SQL Server Integration Service)

SSIS (*SQL Server Integration Services*) adalah suatu platform untuk membangun sistem yang handal untuk integrasi data, *extraction*, *transformation*, dan *loading* yang digunakan pada *data warehousing* [8]. SSIS menawarkan solusi dalam menghadapi permasalahan data integrasi. Selain itu, *tools* ini membatu untuk menigkatkan efisiensi waktu pembuatan.

Arsitektur *SQL Server Integration Services* secara umum berisi berbagai macam komponen, diantaranya yaitu :

1. *SSIS Deginer*. Merupakan *tools* yang digunakan untuk membuat dan mengatur paket *integration service*. Pada *SQL Server 2012*, *tools* ini sudah

terintegrasi dengan *Visual Studio 2010* yang merupakan bagian *project Bussiness Intelegence*.

2. *Runtime Engine*. Komponen ini berguna untuk menjalankan semua paket SSIS yang sudah dibuat.
3. *Task* dan *executable binary*.
4. *Data Flow Engine* dan *Data Flow*. Komponen *data flow* merupakan enkapsulasi *data flow engine* yang menyediakan *buffer* didalam memori dan bertugas memindahkan data dari sumber data ke tujuan data. Sedangkan *data flow* merupakan sumber data, tujuan data, dan transformasi.
5. *Integration Services service*. Memungkinkan *SQL Server Management Studio* dapat digunakan untuk melakukan monitoring paket SSIS dan mengatur SSIS *storage* yang digunakan.
6. *SQL Server Import* dan *Export Wizard*. *Tools* ini digunakan untuk melakukan *copy* data dari sumber ke tujuan data.

2.2.3. UML (Unified Modeling Language)

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam suatu entitas. Pengrtian berorientasi objek berarti bahwa cara mengorganisasikan perangkat lunak sebagai kumpulan dari objek tertentu yang memilki struktur dan perilakunya.

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu :

1. Analisis berorientasi objek (OOA)

   Metode analisis yang memeriksa kebutuhan yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam lingkup perusahaan. UML adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun, dan mendokumentasi yang dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan

sistem non perangkat lunak lainnya. UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan prinsip. Bahasa pemodelan merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Beberapa bagian dari UML adalah sebagai berikut

a) Use Case Diagram

   Menggambarkan sejumlah eksternal dan hubungannya ke *use case* yang diberikan oleh sistem.

b) Activity Diagram

   Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case*.

c) Class Diagram

   Menggambarkan struktur dan deskripsi *class*, *package* (paket), dan objek beserta hubungan satu sama lain seperti *containment* (penahanan), pewarisan, asosiasi dan lain-lain.

d) Sequence Diagram

   Menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem [9].

2. Desain berorientasi onjek (OOD)

   Metode untuk mengarahkan arsitektur perangkat lunak yang didasarkan pada manipulasi objek-objek sistem atau subsistem.