BAB 2
LANDASAN TEORI

2.1 Profil BPTP Jawa Barat

Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat berdiri pada tanggal 1 April 1994 didasarkan Surat Keputusan Menteri Pertanian RI Nomor: 798/Kpts/OT.210/12/94. BPTP Jawa Barat sebelumnya bernama BPTP Lembang merupakan salah satu Unit Pelaksana Teknis Badan Penelitian dan Pengembangan Pertanian yang secara struktural berada di bawah koordinasi Pusat Penelitian dan Pengembangan Sosial Ekonomi Pertanian.

Berdasarkan Keputusan Menteri Pertanian RI Nomor : 350/Kpts/OT. 210/6/2001 nama BPTP Lembang kemudian diubah menjadi BPTP Jawa Barat. Berdasarkan Peraturan Menteri Nomor: 16/Pemerintah/OT.140/3/ 2006 tentang Organisasi dan Tata Kerja Balai Pengkajian Teknologi Pertanian, BPTP Jawa Barat dalam pelaksanaan tugas sehari-hari dikoordinasikan oleh Kepala Balai Besar Pengkajian dan Pengembangan Teknologi Pertanian, BPTP Jawa Barat dalam pelaksanaan tugas sehari-hari dikoordinasikan oleh Kepala Balai Besar Pengkajian dan Pengembangan Teknologi Pertanian (BBP2TP).

Pembentukan BPTP bertujuan untuk mempercepat alih teknologi pertanian, mendukung pembangunan pertanian dan optialisasi pemanfaatan sumberdaya pertanian wilayah, melalui:

1. Akselerasi adopsi teknologi
2. Mendekatkan pelayanan pengkajian kepada masyarakat
3. Menjaga kesinambungan penelitian pengkajian, dan penyuluhan.

2.1.1 Visi dan Misi BPTP Jawa Barat

Visi dan misi dari Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat adalah sebagai berikut:

1. Visi

   Visi dari Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat saat ini adalah:

"Menjadi lembaga pengkajian teknologi pertanian spesifik lokasi yang terkemuka di Indonesia pada tahun 2015".

2. Misi

Misi dari Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat adalah sebagai berikut:

1. Menghasilkan teknologi unggul berbasis sumberdaya lokal
2. Mempercepat dan memperluas penyebaran dan adopsi teknologi spesifik lokasi
3. Memberikan pelayanan prima informasi dan advokasi teknologi spesifik lokasi secara cepat, tepat, mudah, dan murah
4. Mengembangkan kelembagaan produksi pola manajemen industri.

2.1.2 Logo BPTP Jawa Barat

Logo dari Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat dapat dilihat pada Gambar 2.1 dibawah ini.



Gambar 2.1 ***Logo BPTP Jawa Barat***

Arti logo Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat seperti berikut :

1. Tunas, menggambarkan pengertian biologis daripada seluruh kegiatan yang dikelola oleh Departemen Pertanian. Kecuali manusia, sebagai benda hidup. Warna hijau muda melambangkan pengertian kehidupan.
2. Lingkaran berwarna merah, melambangkan pengertian kesatuan.

3. Lingkaran yang bersudut lima buah berwarna coklat, melambangkan pengertian daripada unsur pelaksana utama (tugas-tugas pokok Departemen Pertanian).

4. Warna dasar baik untuk Panji maupun Vendel serta bentuk lainnya adalah "KUNING EMAS" sebagai lambang kemegahan.

5. Air berwarna biru muda, mempunyai pengertian sebagai lambang keagungan.

2.1.3 Tugas Pokok dan Fungsi

Berdasarkan Peraturan Menteri Pertanian Nomor :16/Pemerintah/OT.140/3/ 2006 , BPTP Jawa Barat mempunyai tugas melaksanakan pengkajian, perakitan dan pengembangan teknologi pertanian tepat guna spesifik lokasi. Dalam melaksanakan tugas BPTP menyelenggarakan fungsi :

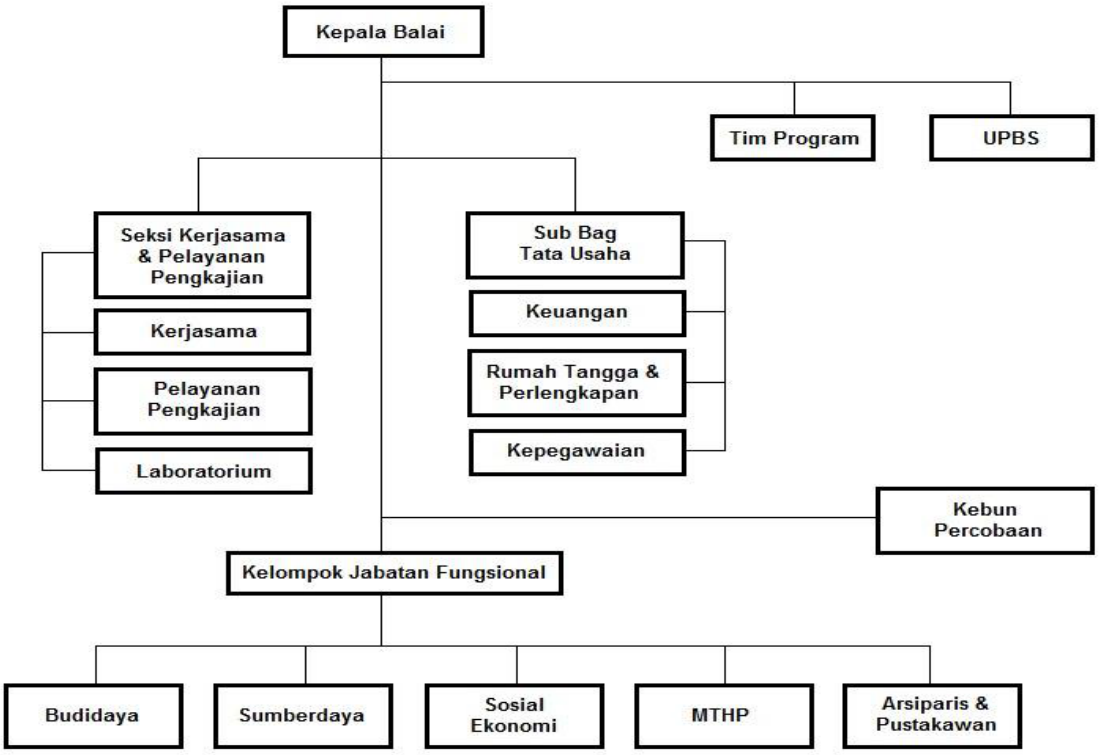
1. Pelaksanaan inventarisasi dan indentifikasi kebutuhan teknologi tepat guna spesifik lokasi;
2. Pelaksanaan penelitian, pengkajian dan perakitan teknologi pertanian tepat guna spesifik lokasi;
3. Pelaksanaan pengembangan teknologi dan diseminasi hasil pengkajian serta perakitan materi penyuluhan;
4. Penyiapan kerjasama, informasi, dokumentasi dan penyebarluasan serta pendayagunaan hasil pengkajian, perakitan, dan pengembangan teknologi pertanian tepat guna spesifik lokasi;
5. Pemberian pelayanan teknik kegiatan pengkajian, perakitan dan pengembangan teknologi pertanian tepat guna spesifik lokasi; dan
6. Pelaksanaan urusan administrasi dan tata usaha balai.

2.1.4 Organisasi

Untuk menjalankan tugas dan fungsinya, BPTP Jawa Barat dipimpin oleh seorang Kepala dengan jabatan struktural eselon IIIa. Dalam pelaksanaan tugas struktural sehari-hari Kepala Balai dibantu Kepala Sub Bagian Tata Usaha dan Kepala Seksi Kerjasama dan Pelayanan Pengkajian dengan jabatan stuktural Iva.

2.1.5 Struktur Organisasi di BPTP Jawa Barat

Susunan organisasi BPTP Jawa Barat berdasarkan Peraturan Menteri Pertanian Nomor : 16/Pemerintah/OT.140/3/2006 terdiri dari: (a) Subbagian Tata Usaha; (b) Seksi Kerjasama dan Pelayanan Pengkajian; dan (c) Kelompok Jabatan Fungsional. Sususan organisasi BPTP Jawa Barat dapat dilihat pada gambar Gambar 2.2 dibawah ini.



Gambar 2.2 Struktur Organisasi BPTP Jawa Barat

Subbagian Tata Usaha mempunyai tugas melakukan urusan kepegawaian, keuangan, perlengkapan, surat menyurat, dan kearsipan, serta rumah tangga. Sedangkan Seksi Kerjasama dan Pelayanan Pengkajian mempunyai tugas melakukan penyiapan bahan penyusunan rencana, program, anggaran, pemantauan, dan evaluasi serta laporan, dan penyiapan bahan kerja sama, informasi, dokumentasi, dan penyebarluasan dan pendayagunaan hasil, serta pelayanan sarana pengkajian, perakitan, dan pengembangan teknologi pertanian tepat guna spesifik lokasi.

2.2 Landasan Teori

2.2.1 Konsep Basis Data

2.2.1.1 Data

Menurut Joseph M., data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk jamak dari datum, berasal dari bahasa Latin yang berarti "sesuatu yang diberikan". Dalam penggunaan sehari-hari data berarti suatu pernyataan yang diterima secara apa adanya [4].

Sedangkan menurut Margy Ross, data merupakan bahasa, simbol-simbol pengganti lain yang disepakati oleh umum dalam menggambarkan objek, manusia, peristiwa, aktivitas, konsep. Singkatnya data merupakan suatu kenyataan apa adanya [5].

Data adalah fakta mentah mengenai orang, tempat, kejadian, dan hal-hal yang penting dalam organisasi [4].

Sehingga kesimpulannya data merupakan rekaman mengenai fenomena yang ada dan yang terjadi. Data pada pokoknya adalah refleksi fakta yang ada.

2.2.1.2 Basis Data

Basis data secara universal disebutkan sebagai sebuah kumpulan data yang disimpan dalam satu tempat yang sama. S. Attre mengatakan bahwa *Database* adalah koleksi data-data yang saling berhubungan mengenai suatu organisasi/*enterprise* dengan macam-macam pemakaiannya [4]. Dalam referensi lainnya, *Database* atau basis data adalah koleksi "data operasional" yang tersimpan dan dipakai oleh sistem aplikasi dari suatu organisasi. Dan jika dijabarkan basis data dikelompokkan menjadi tiga bagian, yakni :

1. Data *input* adalah data yang masuk dari luar sistem.
2. Data *output* adalah data yang dihasilkan sistem.
3. Data operasional adalah data yang tersimpan pada sistem

Menurut Toni Fabbri: *Database* adalah sebuah sistem *file-file* yang terintegrasi yang mempunyai minimal *primary key* untuk pengulangan data.

Menurut Gordon C. Everest: *Database* adalah koleksi atau kumpulan data yang mekanis, terbagi/*shared,* terdefinisi secara formal dan dikontrol terpusat pada organisasi.

2.2.2 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 [6]. 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 [6] 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 [7] 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 [7] 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* [8]. 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* [9]

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

2.2.2.1 Tujuan Data Warehouse

Tujuan dari data *warehouse* adalah sebagai berikut [9] :

1. Memberikan kemudahan untuk mengakses informasi yang ada.

   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.
2. Menyediakan informasi yang konsisten.

   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 dipertanggung jawabkan.
3. Mampu beradaptasi dan tahan terhadap perubahan.

   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.

4. Mampu mengamankan informasi.

   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.

5. Mampu memberikan dukungan dalam pengambilan keputusan.

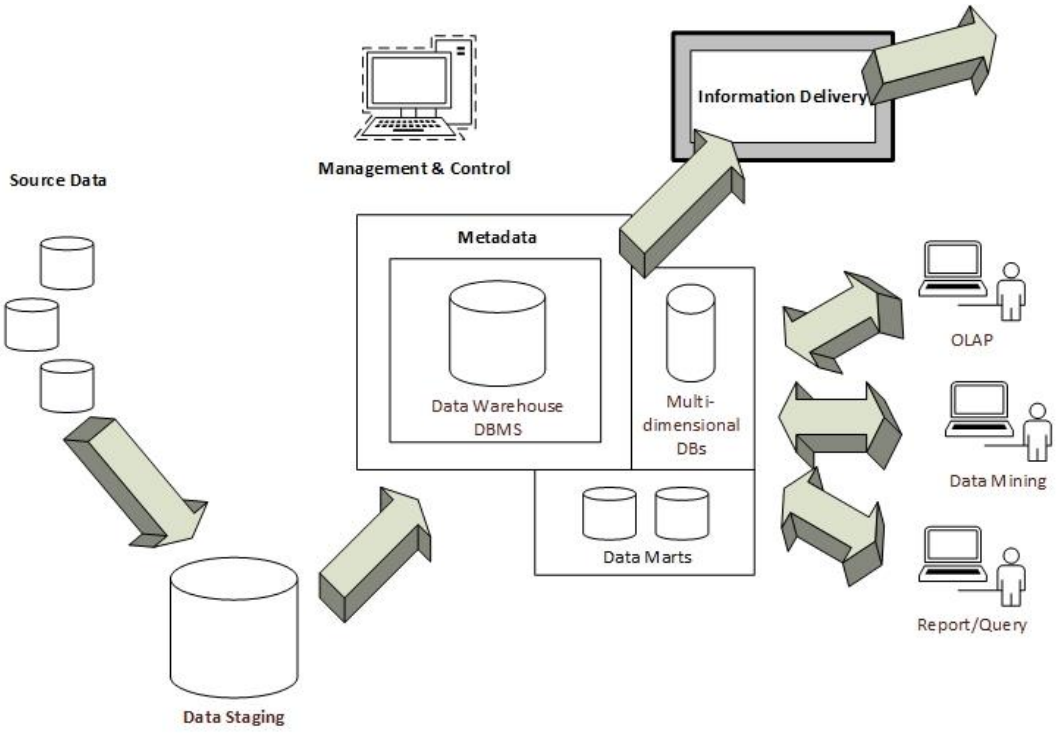
   Ini merupakan tujuan yang paling penting dan harus ada dalam setiap pembuatan data *warehouse*. Data warehouse bisa digambarkan sebagai kumpulan teknologi pendukung keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan informasi, dapat membuat keputusan dengan cepat dan tepat.

6. *User friendly*.

   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.

2.2.2.2 Arsitektur Data Warehouse

Arsitektur pada data *warehouse* dikelompokkan menjadi 5 bagian seperti Gambar 2.3 [10] :



Gambar 2.3 Arsitektur Data *Warehouse* [10]

1. *Source* Data

   Merupakan sumber data, atau bisa dibilang bahwa darimana data itu berasal. Untuk membangun data *warehouse* Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Barat 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.

2.2.2.3 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 [6].

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 [6]. Tabel dimensi umumnya berisi data keterangan, dimana data tersebut jarang sekali mengalami perubahan.

2. Tabel Fakta (*Fact Tables*)

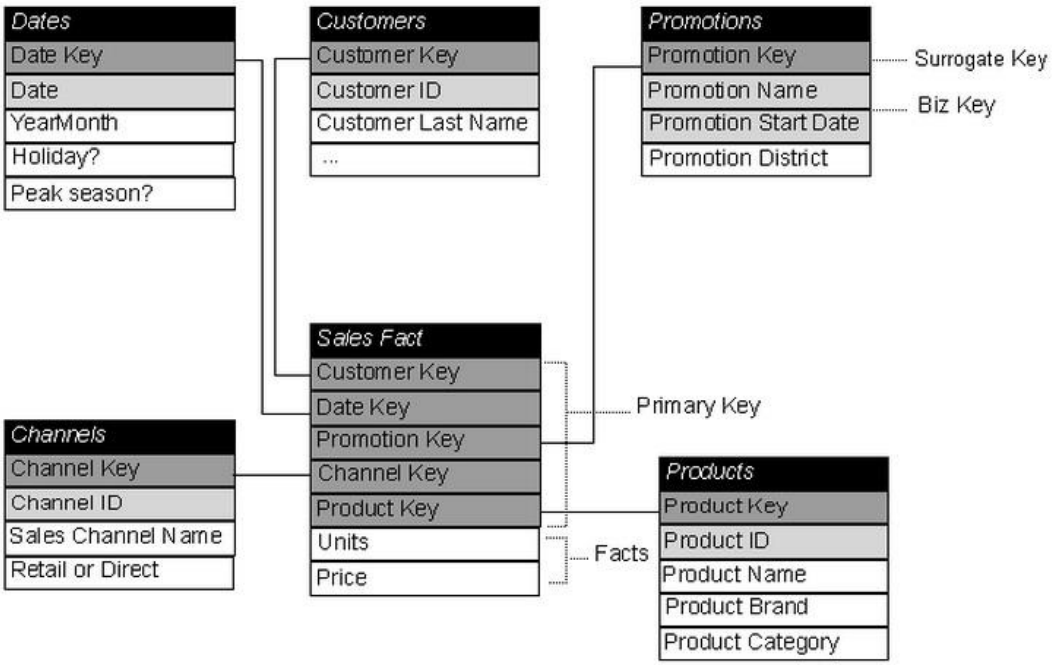
   Tabel fakta merupakan sebuah tabel yang menjelaskan tentang transaksi bisnis dari suatu enterprise biasanya disebut tabel detail [6]. Tabel fakta umumnya berisi data yang berkaitan langsung dengan proses bisnisnya.

3. Skema *Dimensional* Model

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

a) Skema *star*

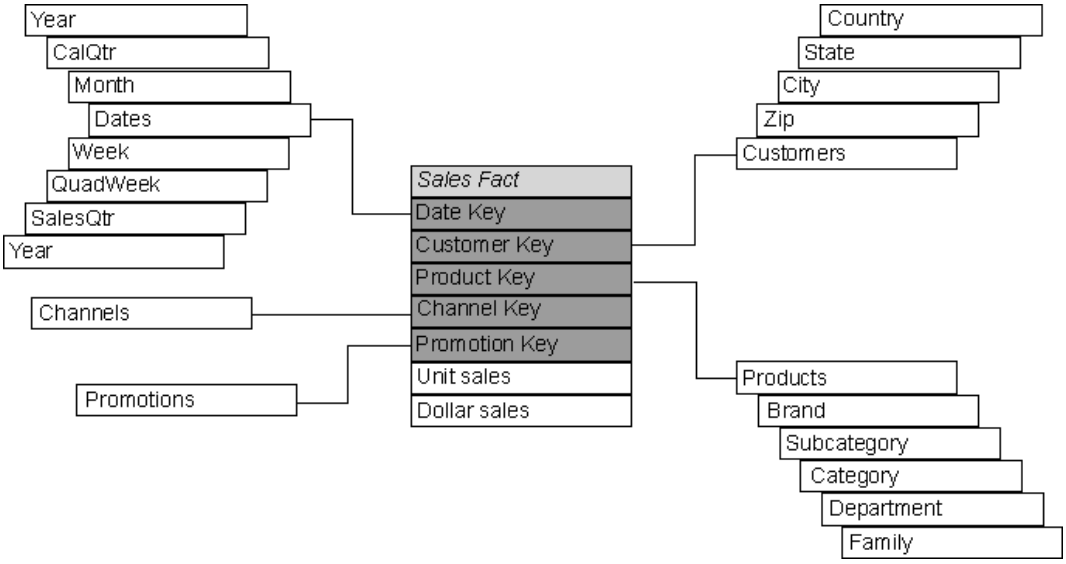
   Suatu skema disebut skema *star* jika seluruh tabel dimensi dihubungkan secara langsung ke tabel fakta dan satu tabel fakta wajib memiliki relasi minimal dengan satu tabel dimensi [6]. Gambar 2.4 dari skema *star*.



Gambar 2.4 Skema *Star* [7]

b) Skema *snowflake*

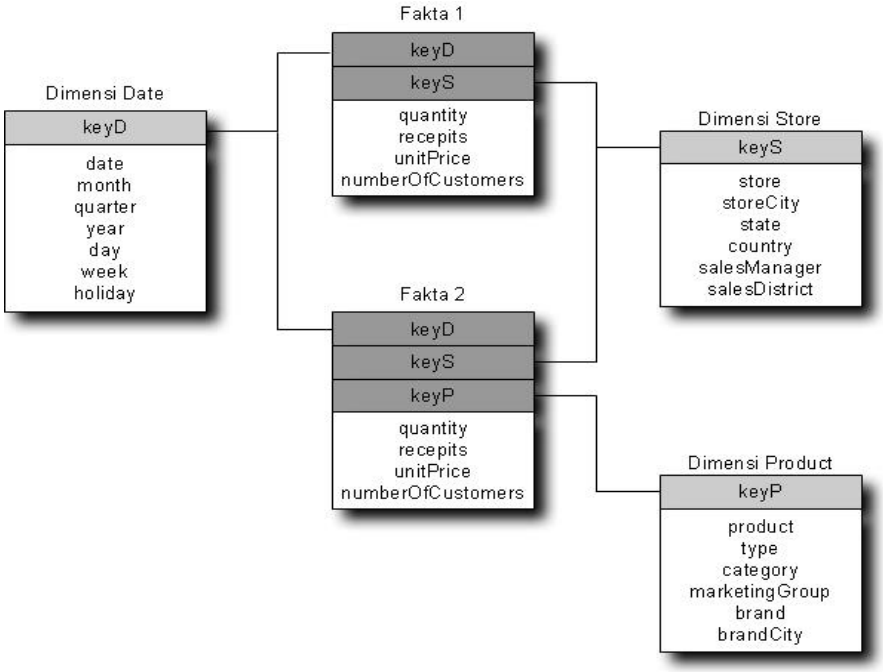
   Suatu skema disebut skema *snowflake* jika satu atau lebih tabel dimensi tidak berhubungan langsung dengan tabel fakta tetapi harus berhubungan melalui tabel dimensi lain [6]. Gambar 2.5 dari skema *snowflake*.



Gambar 2.5 Skema *Snowflake* [7]

c) Skema c*onstellation*

Suatu skema dikatakan sebagai skema *constellation* jika ada satu tabel dimensi yang dipakai bersamaan oleh satu atau lebih tabel fakta [11]. Gambar 2.6 dari skema *constellation.*

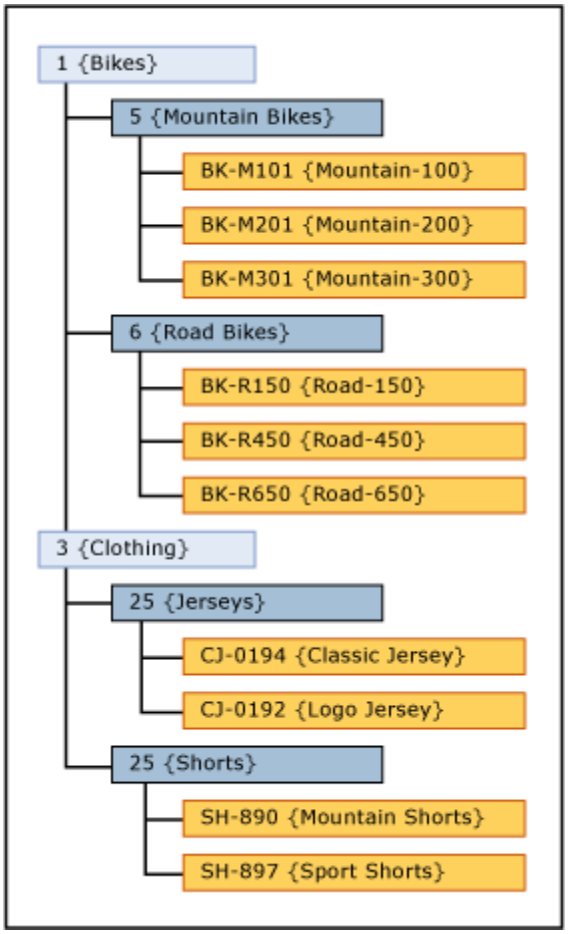


Gambar 2.6 Skema *Constellation* [11]

2.2.2.4 Hirarki (Hierarchies)

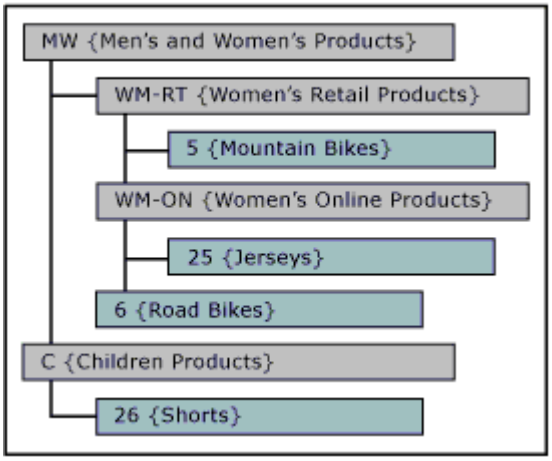
Hirarki merepresentasikan *relationships* antar tiap entitas ataupun tiap anggota entitasnya. Hirarki bisa juga dikatakan sebagai *parent-child relationships* karena hirarki memandang sesuatu melalui berbagai sudut pandang. Hirarki dibagi menjadi:

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 berikut.



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*

2.2.2.5 Proses ETL 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* [12].

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* [12].

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* [10]. 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 *warehouse*. Operasi *loading* terdiri dari memasukkan *record* ke dalam bermacam-macam dimensi dan *fact* tabel dari data *warehouse* [9].

2.2.2.6 Data Warehouse Tools

Berikut ini adalah *tools* yang digunakan *user* setelah data *warehouse* terbentuk dengan tujuan yang berbeda-beda [7] :

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* [7]. 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 [10]. Berikut ini adalah karakteristik paling mendasar dalam OLAP *systems* [10] :

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 [10] 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 [9]:

a) *Consolidation* (*roll-up*)

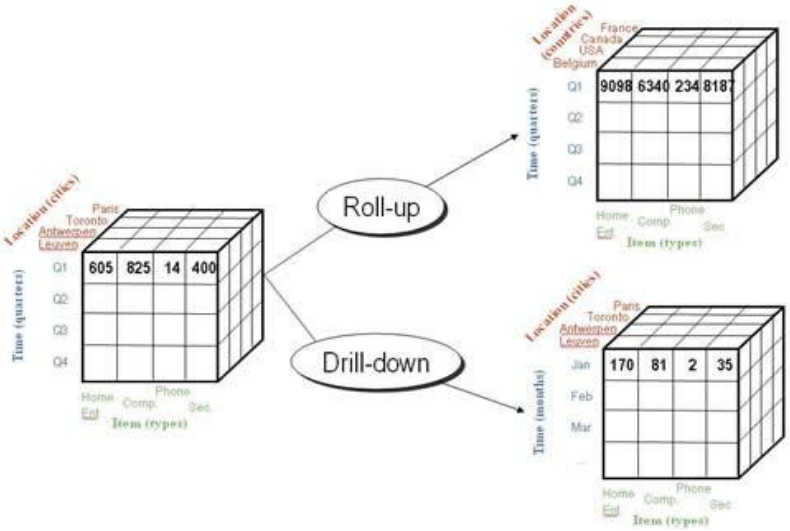
   Konsolidasi melibatkan pengelompokkan data untuk melihat data secara global atau rangkuman (*summary*).

b) *Drill-down*

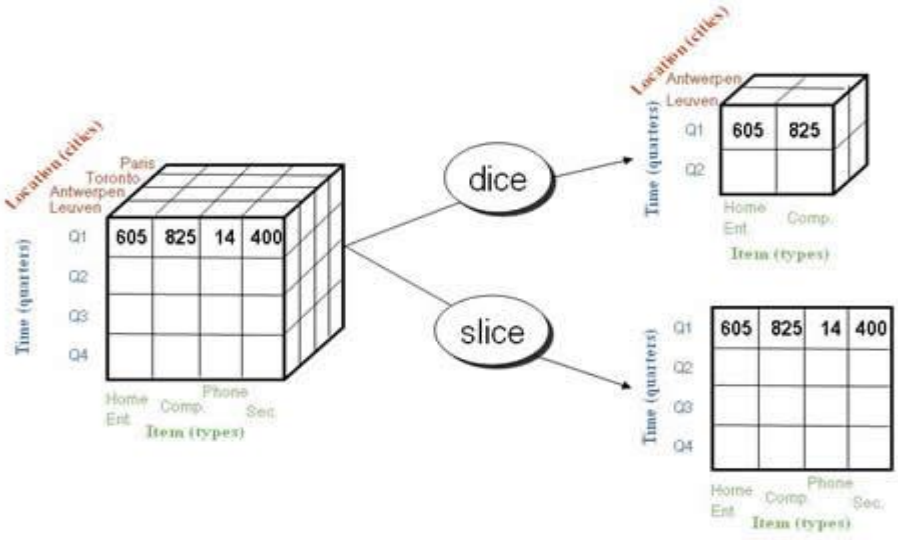
   Suatu bentuk yang merupakan kebalikan dari konsolidasi, untuk mendapatkan lebih detail tentang suatu dimensi serta bisa dikatakan sebagai suatu navigasi dari tingkat yang lebih umum ke tingkat yang lebih spesifik. Untuk lebih jelasnya *roll-up* dan *drill-down* dapat dilihat pada Gambar 2.9.

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



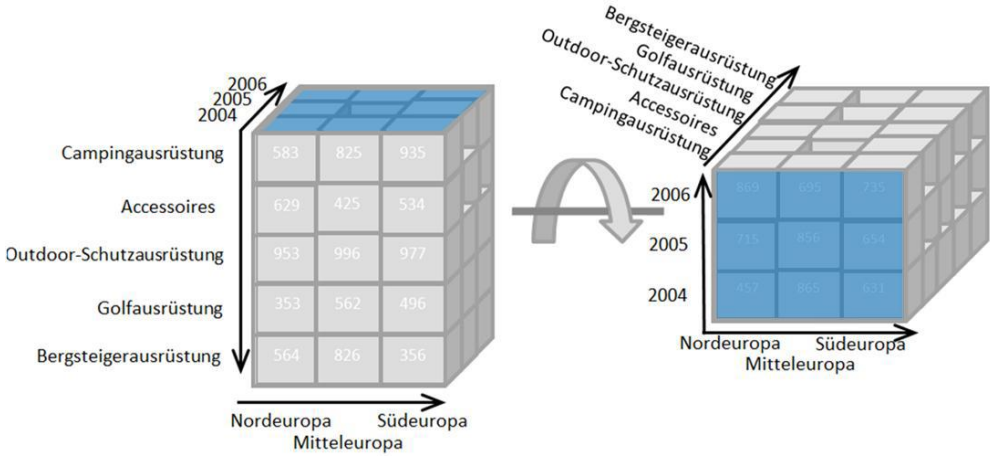
Gambar 2.9 *Roll-up* dan *Drill-down*



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* [9]. OLAP *cube* contohnya seperti pada dan

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 [10]. 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) [7].

2.2.3 Object-Oriented Analysis & Design (OOAD)

Object-Oriented Analysis & Design (OOAD) merupakan salah bentuk pendekatan analisis dan desain yang menekankan solusi berbasis objek [13]. OOAD bukan dipilih berdasarkan bahasa pemrograman yang digunakan melainkan cara pikir kita dalam menyelesaikan masalah dengan menitikberatkan kepada perekayasaan objek beserta relasinya.

Alat bantu yang biasa digunakan untuk membantu pemodelan OOAD adalah UML (*Unified Modeling Languages*). Berikut ini adalah diagram-diagram yang biasa digunakan UML :

1. *Use Case* Diagram

   Menggambarkan sistem berdasarkan hasil *requirement gathering*. Setiap *use case* melambangkan satu fungsional sistem.
2. *Activity* Diagram

   Menggambarkan secara jelas alur kegiatan dari *use case*.
3. *Sequence* Diagram

   Menggambarkan hubungan antar *class* atau bisa dibilang aktivitas apa yang berlangsung sejak objek pertama kali dibuat sampai memenuhi tugas dari objeknya.
4. *Class* Diagram

   Menunjukkan atau memperjelas *class* apa saja yang berada dalam domain permasalahan beserta relasinya.
5. *Statechart* Diagram

   Menunjukkan transisi antar keadaan yang berlangsung.