

BAB II

LANDASAN TEORI

2.1 Konsep Dasar *Data Warehouse*

2.1.1 Pengertian Data

Menurut Deppi Linda (2015), data merupakan kenyataan yang menggambarkan suatu kejadian atau kesatuan nyata.

Menurut Anhar (2017), Data merupakan kenyataan yang menggambarkan suatu kejadian dan merupakan kesatuan nyata yang nantinya digunakan sebagai bahan dasar suatu informasi.

Berdasarkan pengertian dari berbagai ahli di atas, dapat disimpulkan bahwa data adalah fakta-fakta yang disimpan dalam bentuk *file* dimana data tersebut belum diolah tetapi disimpan untuk memberikan arti tertentu kepada orang yang membutuhkan data tersebut.

2.1.2 Pengertian Informasi

Menurut Deppi Linda (2015), informasi merupakan hasil pengolahan dari data. Kualitas dari sebuah informasi bergantung pada keakuratan, ketepatan waktu, dan relevansi. Dengan kebutuhan informasi.

Menurut Rainer & Turban (2016), informasi mengacu pada data yang telah diorganisir sehingga mereka memiliki makna dan nilai kepada penerima. Karakteristik penting yang harus dimiliki oleh suatu informasi adalah:

a. Relevansi

Informasi tersebut berkaitan dengan keputusan yang diambil dalam usaha mencapai tujuan yang telah ditetapkan.

b. Akurat

Informasi dapat diandalkan dan disajikan secara tepat.

c. Tepat Waktu

Informasi harus dapat diterima oleh penerima, dan tidak boleh terlambat karena informasi yang terlambat menjadi tidak bernilai.

d. Kelengkapan

Informasi harus mampu menyajikan gambaran lengkap dari suatu permasalahan atau penyelesaian.

Berdasarkan pengertian dari berbagai ahli di atas, dapat disimpulkan bahwa Informasi merupakan kesimpulan dari kumpulan data yang telah dikelola dengan tujuan dan memiliki karakteristik untuk menghasilkan sesuatu yang bernilai dan berguna untuk penggunaannya.

2.1.3 Pengertian *Database*

Menurut Khusnul Khotimah & Sriyanto (2016), basis data adalah sebuah kumpulan data yang secara logis terkait dan dirancang untuk memenuhi suatu kebutuhan informasi dari sebuah organisasi.

Menurut Indrajani (2015), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

Adapun beberapa istilah penting dalam basis data menurut Connolly & Begg (2015), yaitu:

a. Relasi

Adalah sebuah tabel dengan kolom dan baris.

b. Atribut

Adalah sebuah kolom yang diberi nama pada sebuah relasi.

c. *Relationship*

Adalah asosiasi antar tabel.

d. *Primary key*

Adalah suatu *candidate key* yang terpilih untuk mengidentifikasi *tuple* secara unik dalam sebuah relasi.

e. *Multiplicity*

Adalah jumlah kejadian yang mungkin dari sebuah tipe entitas yang berhubungan dengan kejadian tunggal dari tipe entitas lain yang berhubungan melalui *relationship* tertentu.

Berdasarkan pengertian diatas dapat diambil kesimpulan bahwa sebuah *database* adalah kumpulan data-data yang saling terhubung secara logika satu sama lain dan tersimpan dalam skema tertentu yang digunakan untuk memenuhi kebutuhan informasi yang dibutuhkan oleh pengguna.

2.1.4 Pengertian *Database Management System* (DBMS)

Menurut Connolly & Begg (2015), DBMS adalah sebuah sistem perangkat lunak yang mengizinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke dalam basis data. Fasilitas yang disediakan oleh DBMS diantaranya:

- a. Mengizinkan pengguna untuk mendefinisikan basis data, dengan melalui *Data Definition Language* (DDL). DDL mengizinkan pengguna untuk menentukan tipe, struktur, serta kendala data yang nantinya disimpan ke dalam basis data.
- b. Mengizinkan pengguna untuk melakukan menambah, mengubah, menghapus dan mengambil data dari basis data tersebut, dengan menggunakan *Data Manipulation Language* (DML). Standar bahasa dari DBMS ialah *Structured Query Language* (SQL).
- c. Menyediakan akses kontrol ke dalam basis data, seperti:
 1. Sistem keamanan, yang dapat mencegah pengguna yang tidak diberi kuasa untuk mengakses basis data.
 2. Sistem integritas, yang dapat menjaga konsistensi dari data yang tersimpan.
 3. Sistem kontrol konkurensi, yang mengizinkan berbagi akses dengan basis data.
 4. Sistem kontrol pemulihan, jika terjadi kegagalan perangkat keras atau perangkat lunak maka sistem kontrol pemulihan ini dapat

mengembalikan basis data ke keadaan yang konsisten dari yang sebelumnya.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa *Database Management System (DBMS)* adalah sebuah *software* yang digunakan untuk membangun dan mengelola data dalam *database*.

2.1.5 Entity Relationship Diagram (ERD)

Menurut Sutanta (2015), *Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek.

a. Komponen ERD

Sutanta (2015) juga menjabarkan komponen *Entity Relationship Diagram* adalah sebagai berikut:

1. Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan entitas digunakan aturan sebagai berikut:

- a. Entitas dinyatakan dengan simbol persegi panjang.
- b. Nama entitas dituliskan didalam simbol persegi panjang.
- c. Nama entitas berupa kata benda, tunggal.
- d. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

1. Atribut digambarkan dengan simbol elips.
2. Nama atribut dituliskan didalam simbol elips.
3. Nama atribut merupakan kata benda, tunggal.

4. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

3. Relasi

Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut:

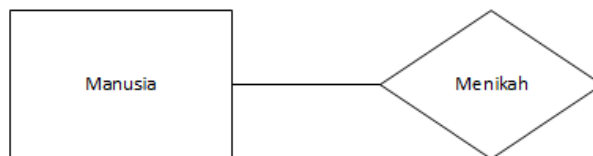
1. Relasi dinyatakan dengan simbol belah ketupat.
2. Nama relasi berupa kata kerja aktif.
3. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

b. Derajat *Relationship*

Derajat *relationship* yang menjelaskan jumlah *entity* yang bervariasi dalam suatu *relationship*. Dalam derajat *relationship* terbagi dari tiga tingkatan yaitu:

1. *Unary Degree* (Derajat Satu)

Unary atau derajat satu adalah satu buah *relationship* menghubungkan satu buah *entity*.



Gambar 2.1 Contoh *Unary Degree*

Keterangan: Manusia menikah dengan manusia, *relationship* menikah hanya menghubungkan *entity* manusia.

2. *Binary Degree* (Derajat Dua)

Binary Degree atau derajat dua adalah satu buah *relationship* yang menghubungkan dua buah *entity*.

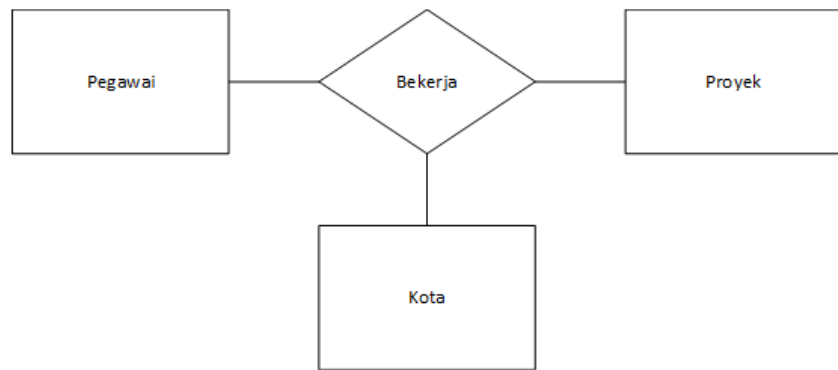


Gambar 2.2 Contoh *Binary Degree*

Keterangan: Pegawai memiliki kendaraan, sebuah *relationship* memilih menghubungkan menghubungkan *entity* Pegawai dan *entity* Kendaraan.

3. Ternary Degree (Derajat Tiga)

Ternary Degree (derajat tiga) adalah satu buah *relationship* menghubungkan tiga buah *entity*.



Gambar 2.3 Contoh *Ternary Degree*

Keterangan: Pegawai pada kota tertentu mempunyai suatu proyek.
Entity Bekerja menghubungkan *Entity* Pegawai, Proyek, dan Kota.

c. Cardinality Ratio Constraint

Menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

1. One to One (1 : 1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding satu.



Gambar 2.4 Contoh *Cardinality One to One*

Keterangan: Satu pasien ditempatkan pada satu kendaraan atau satu kendaraan ditempati satu pasien.

2. *One to Many* (1 : M)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding banyak.



Gambar 2.5 Contoh *Cardinality One to Many*

Keterangan: Satu dokter memeriksa banyak pasien atau banyak pasien diperiksa satu dokter.

3. *Many to One* (M : 1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding satu.

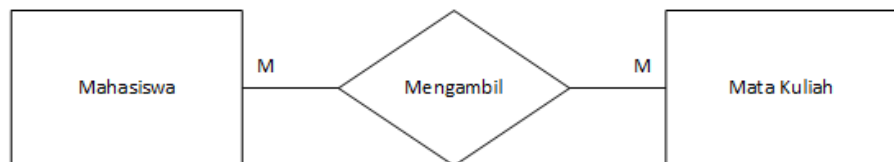


Gambar 2.6 Contoh *Cardinality Many to One*

Keterangan: Banyak pegawai bekerja pada satu departemen atau satu departemen dipekerjakan banyak pegawai.

4. *Many to Many* (M : M)

Yaitu perbandingan antar *entity* pertama dengan *entity* kedua berbanding banyak berbanding banyak.



Gambar 2.7 Contoh *Cardinality Many to Many*

Keterangan: Banyak mahasiswa mengambil banyak mata kuliah atau banyak mata kuliah diambil banyak mahasiswa.

2.1.6 Pengertian *Data Warehouse*

Menurut Kimball (2015), *data warehouse* adalah suatu sistem yang mengekstrak, membersihkan, menyesuaikan diri, dan mengirim sumber data ke dalam suatu data penyimpanan dimensional dan kemudian dapat mendukung implementasi *query* dan analisis untuk tujuan pengambilan keputusan.

Menurut Deppi Linda (2015), *data warehouse* adalah basis data dengan perangkat pelaporan dan *query* yang menyimpan data saat ini dan data historis yang berasal dari beragam sistem operasional dan menggabungkannya untuk pelaporan dan analisis manajemen.

Menurut Inmon (2015), *data warehouse* adalah kumpulan data yang berorientasi subjek, terintegrasi, bervariasi waktu, dan *non-volatile*, yang mendukung proses pengambilan keputusan oleh manajemen.

Jadi dapat disimpulkan bahwa *data warehouse* adalah sekumpulan data yang besar, yang didapatkan perusahaan dari berbagai sumber dalam jangka waktu tertentu, sehingga dapat mendukung pengambilan keputusan oleh pihak manajemen.

2.1.7 Karakteristik *Data Warehouse*

Menurut Han & Kamber (2015) Terdapat empat karakteristik pada *data warehouse*, diantaranya adalah:

a. Subject Oriented

Subject-oriented yaitu data diorganisir berdasarkan subyek utama seperti pelanggan, pemasok, produk, dan penjualan. Dibanding dengan berkonsentrasi pada operasi serta proses transaksi sehari hari dalam sebuah organisasi, *data warehouse* fokus pada pemodelan dan analisis data untuk para pengambil keputusan. Sehingga *data warehouse* memberikan pandangan sederhana terhadap subyek permasalahan dengan tidak berfokus pada data-data yang tidak berguna dalam pengambilan keputusan.

b. *Integrated*

Integrated yaitu *data warehouse* biasanya dibangun dengan mengintegrasikan beberapa *sources* yang berbeda, seperti *database* relasional, *file flat*, dan catatan transaksi *online*. Teknik pembersihan dan integrasi data diterapkan untuk memastikan konsistensi dalam konvensi penamaan, struktur pengkodean, ukuran atribut, dan sebagainya.

c. *Time Variant*

Time Variant yaitu data disimpan untuk menyediakan informasi dalam jangka waktu tertentu misalnya data 5-10 tahun terakhir. Jadi setiap struktur utama dalam *data warehouse* berisi elemen waktu, baik secara implisit atau eksplisit.

d. *Non-Volatile*

Non-volatile artinya sebuah *data warehouse* biasanya terpisah secara fisik yang ditransformasikan dari data aplikasi yang diperoleh di lingkungan operasional. Dengan pemisahan ini, *data warehouse* tidak memerlukan pemrosesan transaksi, pemulihan, dan mekanisme kontrol konkurensi.

2.1.8 **Arsitektur Data Warehouse**

Berdasarkan Connolly & Begg (2015), terdapat komponen-komponen utama didalam arsitektur *data warehouse*, yaitu terdiri dari:

- a. *Operational Data*, sumber dari data ini didapat dari data operasional yang dilakukan pada *database* awal.
- b. *Operational Data Store (ODS)*, tempat penyimpanan data operasional yang sedang terjadi dan terintegrasi dimana digunakan untuk analisis.
- c. *Load Manager* (sering juga disebut komponen *frontend*), menampilkan semua operasi yang diasosiasikan dengan data yang telah diekstrak dan di-load kedalam *data warehouse*.
- d. *Warehouse Manager*, menampilkan semua operasi yang diasosiasikan dengan manajemen data dalam *data warehouse*.
- e. *Query Manager* (sering disebut juga komponen *backend*), menampilkan semua operasi yang diasosiasikan dengan manajemen dari *user queries*.

- f. *Detailed Data*, area dari *data warehouse* yang menyimpan semua data rinci didalam skema *database*. Umumnya, data rinci tidak disimpan secara *online* namun bisa tersedia dengan mengagregasi data ke tingkat detil selanjutnya. Data rinci ditambahkan ke dalam *data warehouse* untuk mendukung data agregat.
- g. *Lightly and Highly Summarized Data*, area dari *data warehouse* yang menyimpan semua standar untuk *lightly summarized* data yang dihasilkan oleh *warehouse manager*. Tujuan dari meringkas informasi adalah untuk mempercepat kinerja saat melakukan *query*. Ringkasan data akan ditambah saat data baru dimasukkan kedalam *data warehouse*.
- h. *Archive Backup Data*, area *warehouse* yang menyimpan *detailed* dan *summarized* data yang bertujuan sebagai arsip dan *backup* data,
- i. *Metadata*, area ini menyimpan definisi *metadata* (data tentang data) yang digunakan oleh semua proses dalam *data warehouse*. *Metadata* digunakan untuk berbagai macam tujuan termasuk:
 1. Proses *extract* dan *load*, *metadata* digunakan untuk memetakan sumber data kedalam pandangan umum sebagai *warehouse*.
 2. Proses manajemen *warehouse*, *metadata* digunakan untuk mengotomatisasi pembuatan tabel ringkasan.
 3. Sebagai bagian proses manajemen *query*, *metadata* digunakan untuk menghubungkan *query* ke sumber data yang sesuai.

Struktur *metadata* berbeda antara setiap proses, karena tujuan berbeda. Ini berarti bahwa berbagai salinan *metadata* yang menggambarkan data *item* yang sama dipegang dalam *data warehouse*. Kebanyakan *tool* penjual untuk salinan manajemen dan *end-user* data mengakses menggunakan versi mereka sendiri dari *metadata*.

- j. *End-User Access Tools*, dapat dikategorikan menjadi lima grup utama, yaitu *data reporting and query tools*, *application development tools*, *executive information system (EIS) tools*, *online analytical processing (OLAP) tools* dan *data mining tools*.

2.1.9 Manfaat *Data Warehouse*

Ada empat manfaat yang bisa dilakukan dengan adanya *data warehouse*, diantaranya yaitu:

1. Pembuatan Laporan

Pembuatan laporan merupakan salah satu manfaat dari *data warehouse* yang paling umum dilakukan. Dengan menggunakan kueri sederhana didapatkan laporan perbulan, pertahun atau jangka waktu kapanpun yang diinginkan.

2. *On-Line Analytical Processing (OLAP)*

Dengan adanya *data warehouse*, semua informasi baik detail maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah didapat. OLAP mendayagunakan konsep data multi dimensi dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL. Hal ini dimungkinkan karena pada konsep multi dimensi, maka data yang berupa fakta yang sama dapat dilihat dengan menggunakan fungsi yang berbeda. Fasilitas lain yang ada pada *software OLAP* adalah fasilitas *roll-up* dan *drill-down*. *Drill-down* adalah kemampuan untuk melihat detail dari suatu informasi dan *roll-up* adalah kebalikannya.

3. Proses Informasi *Executive*

Data warehouse dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan *data warehouse* segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan. Informasi dan data pada laporan *data warehouse* menjadi target *informative* bagi *user*.

2.1.10 Pengertian Denormalisasi

Menurut Connolly & Begg (2015), denormalisasi adalah suatu proses yang merubah bentuk normalisasi dari database dengan cara penggabungan

tabel dan merupakan sebuah proses yang secara sengaja dilakukan dengan melanggar peraturan bentuk normal dari normalisasi dengan tujuan untuk meningkatkan *performance* pengaksesan data yang ada.

Keuntungan melakukan proses denormalisasi adalah:

- a. Mengurangi jumlah relasi yang terjadi antar tabel-tabel yang harus mengalami proses pada waktu pencarian data, sehingga dapat meningkatkan kecepatan proses *query* data.
- b. Membuat struktur fisik *database* agar lebih mudah dipahami menurut model dimensi dari pengguna. Struktur tabel yang dibuat sesuai dengan kebutuhan pengguna memungkinkan terjadinya akses langsung yang juga dapat meningkatkan kinerja.

Kelemahan melakukan proses denormalisasi adalah:

- a. Secara tidak langsung proses denormalisasi menyebabkan penumpukan (*redundancy*) data.
- b. Pada proses denormalisasi memerlukan alokasi penyimpanan yang cukup besar.

2.1.11 Kegunaan *Data Warehouse*

Penerapan *data warehouse* berguna dalam mendapatkan analisa yang lebih baik dari data yang berjumlah sangat besar sehingga dapat membuat keputusan yang baik.

Menurut Connolly & Begg (2015), *data warehouse* yang telah di implementasikan dengan baik dapat memberikan keuntungan bagi perusahaan, yaitu:

- a. Berpotensi mengembalikan nilai yang tinggi atas investasi yang dilakukan perusahaan. Suatu organisasi harus mengeluarkan uang dan sumber daya yang cukup besar untuk memastikan *data warehouse* di implementasikan dengan baik.
- b. Keuntungan kompetitif. Didapatkan jika pengambil keputusan mengakses data bahwa informasi yang sebelumnya tidak tersedia, tidak

diketahui dan tidak dipergunakan misalnya informasi mengenai konsumen, tren, dan permintaan.

- c. Meningkatkan produktifitas para pengambil keputusan perusahaan. *data warehouse* meningkatkan produktifitas para pengambil keputusan perusahaan dengan membuat *database* yang terintegrasi secara konsisten, berorientasi pada subyek, dan data historis.

2.1.12 Langkah Pembuatan *Data Warehouse*

Langkah-langkah yang digunakan dalam melakukan pembuatan *data warehouse* adalah sebagai berikut:

- a. Membaca *data legacy*
Memperhatikan bagian-bagian data yang perlu untuk dilakukan pembersihan.
- b. Memindahkan data dari sumber ke server *data warehouse*
Membuat standarisasi format data lalu meng-*copy* data dari sumber sekaligus membersihkan data yang tidak digunakan.
- c. Memecah *data warehouse* ke dalam tabel fakta dan tabel dimensi
Tabel fakta dan tabel dimensi dibuat dan disusun menurut kebutuhan subyek.

2.1.13 Keuntungan *Data Warehouse*

Menurut Connolly & Begg (2015) terdapat beberapa keuntungan dari *data warehouse* diantaranya adalah:

- a. *Potential High Returns On Investment*
Sebuah organisasi harus menggunakan sejumlah besar sumber daya untuk implementasi *data warehouse* berhasil dan biaya yang bervariasi. Namun, sebuah pembelajaran dari *international data corporation* melaporkan bahwa proyek *data warehouse* memberikan rata-rata *return on investment* (ROI) sebesar 401%.

b. *Competitive Advantage*

Keuntungan bersaing dapat diperoleh dengan mendukung dan memperbolehkan pengambilan keputusan untuk mengakses data yang tidak tersedia, susah didapatkan dan tidak diketahui sebelumnya.

c. *Incrased Productivity of Corporate Decision Makers*

Data warehouse meningkatkan produktivitas dari pengambilan keputusan perusahaan dengan membuat sebuah *database* yang terintegrasi, konsisten, berorientasi subjek dan data historikal. *Data warehouse* mengintegrasikan data dari beberapa sistem menjadi sebuah satu cara pandang yang konsisten bagi organisasi, dengan mengubah data menjadi informasi yang berharga, *data warehouse* mengijinkan pengambilan keputusan dalam perusahaan untuk membuat analisis yang konsisten dan akurat.

Hal terpenting dari *data warehouse* yang dikutip dari Jurnal Ilmiah Widyawati (2017) adalah kemampuan *data warehouse* dalam menyediakan informasi yang dibutuhkan oleh user. *Data warehouse* pemasaran produk ini dapat menyediakan informasi pemasaran produk yang sesuai kebutuhan dengan cara yang mudah dan cepat sehingga dapat mendukung proses evaluasi dan pengambilan keputusan yang dilakukan. Kemampuan yang dimiliki dalam *data warehouse* adalah sebagai berikut:

- a. *Roll-up* dan *drill-down*.
- b. *Output customization*.
- c. *Query* sesuai kebutuhan.

2.1.14 Pengertian *Metadata*

Metadata adalah data mengenai data. *Metadata* memberikan peranan yang penting untuk ke efektifan penggunaan *data warehouse*, karena dapat mempermudah *end user* dalam melakukan analisis dan menghemat waktu. *Metadata* bertindak seperti *index* mengenai isi dari *data warehouse*.

2.1.15 Pengertian OLTP (*Online Transaction Processing*)

Menurut Connolly & Begg (2015), *Online transaction processing* (OLTP) adalah sebuah sistem yang dirancang untuk memaksimalkan kapasitas pemrosesan transaksi.

Berikut ini beberapa keuntungan yang diperoleh dengan menerapkan *Online transaction processing* (OLTP), yaitu:

- a. Efektif dan efisien waktu karena data transaksi disimpan setiap hari, sehingga tidak ada penumpukan data yang belum disimpan.
- b. Data sistematis dalam *data warehouse* karena data disimpan setiap tanggal transaksi.

2.1.16 Pengertian OLAP (*Online Analytical Processing*)

OLAP (*Online Analytical Processing*) biasanya ditandai dengan volume transaksi yang relative rendah. *Query* biasanya sangat rumit dan terdapat agregasi. Untuk sistem OLAP waktu respon adalah sebuah ukuran efektivitas. Aplikasi OLAP banyak digunakan untuk teknik *data mining*. Pada database OLAP ada agregasi, data historis, disimpan dalam skema *multidimensional*.

2.1.17 Perbedaan OLTP dan OLAP

Beberapa perbedaan antara sistem OLTP atau *On-line Transaction Processing System* dengan OLAP atau *On-line Analytical Processing System* yaitu, OLTP merupakan sistem *database online* operasional yang digunakan untuk melakukan sebuah transaksi *online* yang dilakukan dalam hari ke hari dimana di dalam OLTP terjadi proses kueri. Sedangkan *data warehouse* menyediakan kepada *user* atau ahli sebuah aturan dari analisis data dan pembuatan keputusan. Sistem ini dikenal dengan *Online Analytical Processing* atau OLAP. Berikut ini perbedaan sistem OLTP dan OLAP menurut Han & Kamber:

Tabel 2.1 Perbedaan OLTP dan OLAP

	OLTP	OLAP
Karakteristik	Proses operasional	Pemrosesan informasi
Orientasi	Transaksi	Analisis
Pengguna	DBA, <i>database</i> profesional	Para ahli (<i>manager, analyst</i>)
Fungsi	Operasi sehari hari	Informasi yang lama yang dibutuhkan untuk mendukung pengambilan keputusan
Desain <i>database</i>	Berdasarkan ER relasi entitas, berorientasi pada aplikasi	Berdasarkan <i>schema</i> , berorientasi objek
Data	Data yang digunakan adalah data sekarang, data terjamin pada masalah <i>up-to-date</i>	Data historis, perbaikan akurasi data dari waktu ke waktu
<i>Summarization</i>	Data primitif, sangat detail	Gabungan, ringkasan
<i>View</i> (gambaran)	Detail, relasi datar	<i>Summarized</i> (peringkasan), multidimensional
Unit kerja	Pendek, transaksi sederhana	Komplek kueri
Akses	<i>Read/write</i>	<i>Read only</i>
Fokus	Data masuk	Informasi keluar
Operasi	<i>Index / hash</i> pada <i>primary key</i>	Kebanyakan <i>scan</i>
Jumlah data yang diakses	Puluhan	Ratusan hingga ribuan
Jumlah pengguna	Ratusan hingga ribuan	Puluhan

Ukuran <i>database</i>	Berkisar 100 MB	100 MB lebih sampai GB
Prioritas	Performa tinggi, ketersediaan tinggi	Fleksibilitas tinggi, otonomi pengguna akhir
Metrik	Melalui transaksi	Melalui kueri, waktu respon

2.1.18 Pengertian *Extract, Transform and Load (ETL)*

Extract, Transform dan Load (ETL) adalah salah satu proses dalam *data warehouse* yang melibatkan pembacaan data dari sumbernya, lalu dilakukan pembersihan dan penyesuaian format dan penulisan data ke ruang penyimpanan untuk dapat digunakan lebih lanjut.

Proses ETL terdiri dari tiga proses, yaitu *extraction, transformation and loading* yang harus dilalui dalam membangun *data warehouse*:

a. *Extraction*

Extraction adalah langkah pertama dalam proses memasukkan data ke dalam *data warehouse*. Membaca dan memahami sumber data dan menyalin data yang dibutuhkan ke dalam sistem ETL untuk memanipulasi kedepannya. Mulai dari saat ini, data merupakan milik *data warehouse*. (Kimball & Ross, 2015)

b. *Transformation*

Transformation adalah tahap yang menggunakan beberapa aturan atau fungsi dari data yang sudah diambil dan mengukur bagaimana data ini dapat digunakan untuk analisis dan transformasi jumlah tersebut dapat mencakup data, pengkodean data, penggabungan data, pemisahan data, perhitungan data, dan pembuatan kunci pengganti. (Connolly & Begg, 2015)

c. *Loading*

Loading adalah tahapan untuk memasukkan data yang sudah melalui proses transformasi ke dalam *data warehouse*. (Connolly & Begg, 2015)

2.2 Model Data Multidimensi

2.2.1 Dimensional Table (Tabel Dimensi)

Menurut Conolly & Begg (2015), *dimensional modeling* adalah teknik *logical design* yang bertujuan untuk menyajikan data standar, bentuk intuitif yang memungkinkan untuk mengakses *high performance*. Disebut juga tabel kecil (*minor table*), biasanya lebih kecil dan memegang data deskriptif yang mencerminkan dimensi suatu bisnis. Tabel dimensi adalah tabel yang berisikan kategori dengan ringkasan data detil yang dapat dilaporkan laporan keuntungan pada tabel fakta dapat dilaporkan sebagai dimensi waktu (yang berupa perbulan, perkuartal, dan pertahun).

Menurut Kimball & Ross (2015), *Dimension* adalah *entity* yang independen dalam model dimensi yang berperan sebagai *entry point* atau mekanisme untuk *slicing and dicing* bahan tambahan yang terletak pada tabel fakta dan *dimensional model*.

Dapat disimpulkan bahwa, *Dimension* adalah kumpulan dari referensi informasi yang dapat diukur, bersifat individual, tidak berulang, berguna sebagai *grouping* dan *labeling*.

Product Dimension Table
Product Key (PK)
Product Description
SKU Number (Natural Key)
Brand Description
Category Description
Department Description
Package Type Description
Package Size
Fat Content Description
Diet Type Description
Weight
Weight Units of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
... and many more

Gambar 2.8 Tabel Dimensi (*Dimension Table*)

2.2.2 *Cube, Dimension, Measure, dan Member*

Pengertian dari *cube, dimension, measure* dan *member*, yaitu:

- a. *Cube* adalah struktur data multi dimensi konseptual, yang terdiri dari *dimension* dan *measure* dan biasanya mencakup pandangan bisnis tertentu seperti penjualan, *profit*, dan lain sebagainya.
- b. *Dimension* atau dimensi adalah struktur *view* atau sudut pandang yang menyusun *cube*. Dimensi dapat terdiri dari berbagai *level*.
- c. *Measure* adalah nilai pengukuran.
- d. *Member* adalah isi atau anggota dari suatu *dimensional* atau *measure* tertentu.

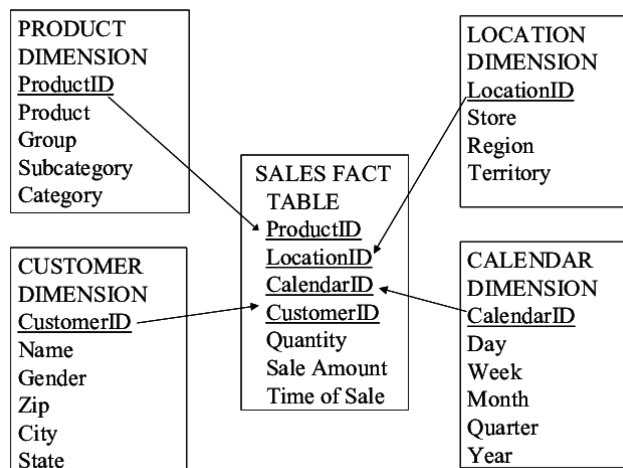
2.2.3 *Surrogate Key*

Surrogate key adalah *key* atau kolom data di tabel dimensi yang menjadi *primary key* dari tabel tersebut. Nilai ini biasanya berupa nilai sekuensial dan biasanya tidak memiliki arti dari proses bisnis darimana sumber data berasal. *Surrogate key* sendiri dibutuhkan karena *data warehouse* tidak hanya berasal dari satu sumber saja, melainkan dari banyak sumber.

2.2.4 **Skema Bintang (Star Schema)**

Menurut Connolly dan Begg (2015), *star schema* adalah model data dimensional yang mempunyai tabel fakta yang berisi data fakta di bagian tengah, dan dikelilingi oleh tabel-tabel dimensi yang terdiri dari data referensi (yang biasanya dapat di denormalisasi). *Star schema* mengambil karakteristik dari data fakta yang di-*generate* oleh *event* yang terjadi di masa lampau.

Menurut Poe (2016), skema bintang adalah metode perancangan yang dilakukan dengan struktur sederhana dengan menggunakan beberapa tabel dan hubungan antar tabel yang jelas. Skema bintang dapat menghasilkan waktu respon yang cepat dalam kueri data dibandingkan dengan proses transaksional dengan menggunakan struktur normalisasi dan memudahkan *end user* untuk memahami struktur basis data pada *data warehouse* yang dirancang.



Gambar 2.9 Skema Bintang (Star Schema)

Skema bintang memiliki beberapa kelebihan yang tidak ada dalam struktur relasional biasa. Keuntungan skema bintang yaitu:

a. Efisiensi

Struktur database konsisten sehingga efisien dalam mengakses data dengan menggunakan *tool* untuk menampilkan data termasuk laporan tertulis dan *query*.

b. Kemampuan untuk Mengatasi Perubahan Kebutuhan

Skema bintang dapat beradaptasi terhadap perubahan kebutuhan pengguna karena semua tabel dimensi memiliki kesamaan dalam hal menyediakan akses ke tabel fakta.

c. *Extensibility*

Model dimensional dapat di kembangkan. Seperti menambah tabel fakta selama data masih konsisten, menambah tabel dimensi selama ada nilai tunggal di tabel dimensi tersebut yang mendefinisikan setiap *record* tabel fakta yang ada, menambahkan *attribute* tabel dimensi, dan memecahkan *record* tabel dimensi yang ada menjadi level yang lebih rendah daripada level sebelumnya.

d. Kemampuan untuk Menggambarkan Situasi Bisnis pada Umumnya

Pendekatan standar untuk menangani situasi umum di dunia bisnis yang terus bertambah.

e. Proses *Query* yang Bisa di Prediksi

Aplikasi *Data Warehouse* yang mencari data dari level yang paling dibawahnya dengan mudah menambahkan jumlah *attribute* pada tabel dimensi dari sebuah skema bintang. Aplikasi yang mencari data dari level yang setara akan menghubungkan tabel fakta yang terpisah melalui tabel dimensi yang dapat diakses bersama.

Selain beberapa kelebihan dari skema bintang diatas, ada beberapa kekurangan bila menggunakan skema bintang, diantaranya adalah:

a. Memakan Penyimpanan yang Besar

Pada penyimpanan atau *memory*, skema bintang membutuhkan ukuran penyimpanan yang relatif besar.

b. Perawatan yang Sulit

Dikarenakan tabel yang tidak normal, maka perawatan atau *maintenance* dapat menjadi lebih sulit.

c. *Long Time Loading Dimension Table*

Untuk proses *loading* pada tabel dimensi memakan waktu yang cukup lama, ketika data yang dimiliki rendah dalam integritas dan nilai dalam replika tinggi, maka waktu untuk melakukan *loading* dapat meningkat.

d. Tidak Fleksibel

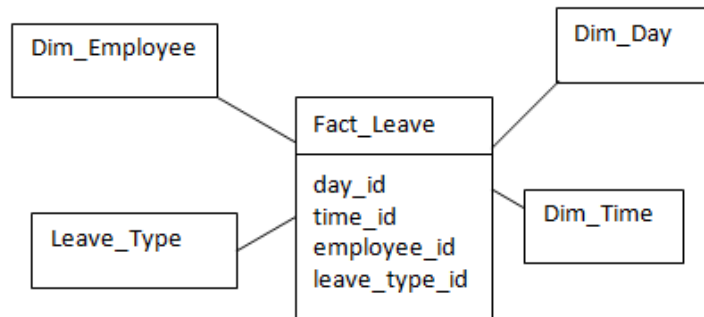
Skema bintang tidak fleksibel dalam kebutuhan analisis, seperti model data yang dinormalisasi (jika data yang dibutuhkan melakukan normalisasi). Skema bintang dibangun hanya untuk kebutuhan data tertentu, sehingga tidak benar-benar memungkinkan analisis yang lebih kompleks.

2.2.5 *Fact Table* (Tabel Fakta)

Menurut Connolly & Begg (2015), tabel fakta adalah tabel pusat dari skema bintang dimana data yang sering muncul dapat di tempatkan disini. Disebut juga tabel utama (*major table*), merupakan inti dari skema bintang dan berisi data aktual yang dianalisis (data kuantitatif dan transaksi). Tabel fakta menyimpan tipe-tipe *measure* yang berbeda, seperti *measure* yang

secara tidak langsung terhubung dengan tabel dimensi dan *measure* yang tidak berhubungan dengan tabel dimensi.

Menurut Kimball & Ross (2015), *Fact Table* adalah table utama dengan performa perhitungan karakteristik *numeric* berdasarkan *composite key*, serta setiap elemen *foreign key* yang diambil dari tabel dimensi.



Gambar 2.10 Tabel Fakta (*Fact Table*)

2.3 Alat Bantu Rancang Bangun *Data Warehouse*

Pada *tools* yang digunakan untuk mengimplementasikan *data warehouse* adalah SQL Server Business Intelligence Development Studio 2008, dalam hal ini adalah: SQL Server Integration Services (SSIS) sebagai *tools* ETL dan SQL Server Analysis Services (SSAS) sebagai OLAP. Berikut diuraikan sebagai berikut:

2.3.1 SQL Server Integration Service (SSIS) sebagai *tool* ETL

SQL Server Integration Services atau biasa disingkat dengan SSIS adalah sebuah *tools* yang digunakan untuk melakukan proses *Extract, Transform, dan Load* (ETL) dan diklasifikasikan sebagai fitur *Business Intelligence* (BI). ETL adalah proses untuk mengumpulkan data dari berbagai sumber (*Extract*), membersihkannya (*Transform*), untuk kemudian menyimpannya ke dalam sistem yang lain (*Load*).

2.3.2 SQL Server Analysis Service (SSAS) sebagai *tool* OLAP

Analysis Services pada SQL Server adalah membicarakan *Business Intelligence* (BI). BI di mulai dari proses pengumpulan data (ETL)

dilanjutkan dengan penyajian data kemudian interpretasi dari data tersebut yang berupa hasil analisa, tentunya semua itu harus di dukung dengan data yang benar, waktu yang tepat dan format yang benar. BI merupakan *business tool* yang efektif untuk mendukung keputusan bisnis, baik berupa keputusan strategik, *tactical* ataupun *operational*.

2.3.3 SQL Server Report Service (SSRS) sebagai *tool Report*

SQL Server Report Services atau biasa disingkat (SSRS) adalah sebuah *tools* yang menyediakan berbagai fungsi dan layanan yang memudahkan dalam membuat, menyebarkan, dan mengelola laporan, serta fitur pemrograman yang memungkinkan untuk memperluas dan menyesuaikan fungsi pelaporan.

Reporting Services adalah pelaporan berbasis *platform server* yang menyediakan fungsi pelaporan yang komprehensif untuk berbagai sumber data. *Reporting Services* mencakup seperangkat lengkap alat untuk membuat, mengelola, dan menyampaikan laporan, dan memungkinkan pengembang untuk mengintegrasikan atau memperpanjang data dan pengolahan laporan dalam aplikasi. *Reporting Services tools* bekerja dalam Microsoft Visual Studio sepenuhnya terintegrasi dengan alat dan komponen SQL Server.

2.4 Fluktuasi Hasil Produksi Pertanian

2.4.1 Pengertian Fluktuasi

Menurut Surya, Yohanes (2017) Fluktuasi adalah perubahan naik atau turunya suatu variabel yang terjadi sebagai akibat dari mekanisme pasar. Secara tradisional fluktuasi dapat diartikan sebagai perubahan nilai. Pengertian fluktuasi adalah lonjakan atau ketidaktetapan segala sesuatu yang bisa digambarkan dalam sebuah grafik. Contohnya seperti fluktuasi harga barang, guncangan atau fluktuasi dalam pengukuran gelombang listrik, dll.

Definisi fluktuasi atau pengertian fluktuasi adalah lonjakan atau ketidaktetapan segala sesuatu yang bisa digambarkan dalam sebuah grafik.

Contohnya seperti fluktuasi harga barang, guncangan atau fluktuasi dalam pengukuran gelombang listrik, dll.

Naik turunnya harga (fluktuasi) dan tingkat harga dari produk-produk pertanian dilihat dari kenyataan-kenyataan yang berlangsung di masyarakat, dengan adanya patokan harga dari pemerintah telah dapat dikendalikan dengan baik, dimana naik dan turunnya itu serta tingkatannya hanya berkisar di antara harga patokan tersebut. Dalam penetapan harga kita banyak belajar dari pengalaman, tata niaga pada masa liberal sering memberi kesempatan pada para produsen untuk mengemukakan harga minimum atas produknya, ini berarti bahwa para pedagang akan mengatur sendiri harga yang akan diterapkan pada produk yang akan diperdagangkannya dengan memperhitungkan pengeluaran-pengeluaran biaya di tingkat grosir dan di tingkat pedagang eceran mengundang persaingan di antara para pedagang.

Berdasarkan pengertian fluktuasi tersebut diatas dapat penulis simpulkan bahwa fluktuasi adalah suatu perubahan variabel tertentu yang umumnya terjadi karena mekanisme pasar. Perubahan itu dapat berupa kenaikan atau penurunan nilai variabel tersebut.

2.4.2 Pengertian Produksi

Produksi adalah kegiatan yang dilakukan manusia dalam menghasilkan suatu produk, baik barang tau jasa yang kemudian dimanfaatkan oleh konsumen.

Menurut Sugiarto (2017), produksi merupakan kegiatan yang merubah input menjadi output. Kegiatan ekonomi ini biasanya dinyatakan dalam fungsi produksi.

Pada saat kebutuhan manusia masih sedikit dan masih sederhana, yaitu seorang memproduksi untuk memenuhi kebutuhannya sendiri. Namun, seiring dengan semakin beragamnya kebutuhan dan keterbatasannya sumber daya, maka seseorang tidak dapat lagi memproduksi apa yang menjadi kebutuhan tersebut.

Secara Umum, Pengertian Produksi merupakan suatu kegiatan atau aktivitas untuk dapat menciptakan/menghasilkan atau juga menambah nilai guna terhadap suatu barang ataupun jasa untuk dapat memenuhi kebutuhan oleh orang ataupun suatu badan (produsen). Orang maupun badan yang melakukan kegiatan atau aktivitas produksi ini dikenal dengan sebutan produsen. Sedangkan untuk barang atau jasa yang dihasilkan dari melakukan kegiatan atau aktivitas produksi disebut dengan sebutan produk. Sedangkan apabila dalam arti ekonomi, Pengertian Produksi ialah sebagai kegiatan atau aktivitas mengenai penciptaan dan juga penambahan atau utilitas terhadap sebuah barang serta jasa.

2.4.3 Fungsi Produksi

Menurut Mankiw (2016), fungsi produksi merupakan hubungan antara jumlah input yang digunakan untuk membuat satu barang dan jumlah output barang tersebut. Kenaikan dalam output produksi yang muncul dari unit tambahan input merupakan produk marginal dan penurunan produk marginal adalah properti dimana produk marginal input menurun ditandai dengan jumlah input meningkat.

Fungsi produksi juga dapat diartikan suatu hubungan yang ketergantungan antara tingkat input yang digunakan dalam proses produksi dengan tingkat output yang dihasilkan dari proses produksi.

2.4.4 Pengertian Pertanian

Menurut Rodjak (2015), petani sebagai unsur usaha tani memegang peranan yang penting dalam pemeliharaan tanaman agar dapat tumbuh dengan baik, ia berperan sebagai pengelola usaha tani. Pertanian adalah suatu jenis kegiatan produksi yang berlandaskan pada proses pertumbuhan dari tumbuh-tumbuhan.

Menurut Soetriono (2016), pertanian dalam arti sempit dinamakan dengan pertanian rakyat, sedangkan pertanian dalam arti luas meliputi pertanian dalam arti sempit, kehutanan, peternakan dan perikanan,

merupakan suatu hal yang penting. Pertanian (*agriculture*) tidak hanya merupakan kegiatan ekonomi untuk menghasilkan pendapatan saja. Lebih dari itu, petani adalah sebuah cara hidup bagi sebagian besar petani.

Konsep pertanian tidak akan menjadi suatu kebenaran umum, karena akan selalu terkait dengan paradigma dan nilai budaya petani lokal, yang memiliki kebenaran umum tersendiri. Oleh sebab itu pemikiran sistem agribisnis yang berdasarkan prinsip filsafat sudah saatnya di pertanyakan kembali. Pertanian memiliki paradigma yang akansarat dengan sistem nilai, budaya, dan ideologi yang patut kita kaji kecocokannya untuk diterapkan di negara kita. Masyarakat petani kita memiliki seperangkat pandangan, nilai, dan falsafah terhadap kehidupan (ideologi) mereka sendiri, yang perlu digali dan di anggap sebagai potensi besar di sektor pertanian. Sementara itu perubahan pandangan dari peningkatan produksi ke pandangan peningkatan pendapatan petani belum cukup jika tanpa dilandasi pada peninjauan kesejahteraan petani.