

BAB II

LANDASAN TEORI

2.1 Data

Data adalah sumber informasi yang bentuknya masih mentah. Menurut Jogianto (1990), 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. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa angka, kata-kata, atau citra.

Dalam keilmuan (ilmiah), fakta dikumpulkan menjadi data. Data kemudian diolah sehingga dapat diutarakan secara jelas dan tepat agar dapat dimengerti oleh orang lain yang tidak langsung mengalaminya sendiri, hal ini dinamakan deskripsi.

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai [8].

2.2 Informasi

Informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Informasi diperoleh dari kegiatan pengumpulan data. Setelah terkumpul, data kemudian diolah dan dianalisis serta dipilah-pilah sehingga didapatkan informasi yang berguna dan sesuai dengan kebutuhan. Kualitas dari sebuah informasi ditentukan oleh tiga hal yaitu, keakuratan informasi, ketepatan waktunya (*up-to-date*), dan relevan dengan keadaan atau kebutuhan.

2.3 Basis Data

Basis data adalah suatu koleksi data yang saling berhubungan secara logikal, dan sebuah deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi (Connolly, 2002, p14). Menurut O'Brien (2005, p141), *database* adalah kumpulan data yang terintegrasi dari elemen-elemen data yang terhubung secara logikal.

2.4 Data Warehouse

Menurut O'Brien (2005, p143), data *warehouse* adalah kumpulan data terintegrasi yang diekstrak dari *database* operasional, historikal, dan eksternal, dibersihkan, ditransformasi, dan dikatalogkan untuk penarikan dan analisis (*data mining*), untuk menyediakan *Business Intelligence* untuk pengambilan keputusan bisnis.

Menurut Inmon (2005:29), data *warehouse* adalah sekumpulan data yang bersifat *subject-oriented*, *integrated*, *time-variant* dan *non-volatile* untuk mendukung proses pengambilan keputusan strategis organisasi[3].

Menurut (kimball, ross, 2002) *data warehouse* adalah sebuah sarana yang dapat meningkatkan kualitas pengambilan keputusan, *data warehouse* berisi data yang dapat mendukung proses DSS.

2.4.1 Karakteristik Data Warehouse

Menurut Inmon (2005, p29), karakteristik dari data *warehouse* yaitu: *subject oriented*, *integrated*, *nonvolatile* dan *time variant*.

1. Subject Oriented (Berorientasi Subjek)

Data *warehouse* diorganisasikan di sekitar subjek-subjek utama dari perusahaan atau organisasi dari pada area-area aplikasi utama. Hal ini tercermin dari kebutuhan untuk menyimpan data yang mendukung pengambilan keputusan dari pada data yang berorientasi aplikasi.

2. Integrated (Terintegrasi)

Terintegrasi karena berasal dari sumber data system aplikasi perusahaan yang berbeda. Sumber data seringkali tidak konsisten. Sebagai contoh, format tabel-tabel sumber data berbeda. Sumber data yang terintegrasi harus dibuat konsisten untuk menyajikan tampilan yang seragam kepada *user*.

3. Time Variant (Variansi Waktu)

Data di dalam *warehouse* hanya akurat dan valid pada beberapa titik waktu atau pada interval waktu tertentu. Variansi waktu dari *data warehouse* juga ditunjukkan oleh perpanjangan

waktu yang dimiliki oleh data, dan asosiasi implisit atau eksplisit waktu dengan semua data, serta fakta bahwa data merepresentasikan serangkaian *snapshot*.

4. *Nonvolatile* (Tidak Mudah Berubah)

Data tidak diupdate secara *real-time* tetapi diperbaharui dari sistem operasional pada basis sehari-hari. Data baru selalu ditambahkan sebagai suplemen untuk sebuah *database*, bukan pengganti. *Database* secara terus menerus menyerap data baru, dan secara bertingkat diintegrasikan dengan data sebelumnya.

Hal-hal yang berkaitan dengan *data warehouse* dalam penerapan pada sistem business intelligence (Inmon, 2002) adalah:

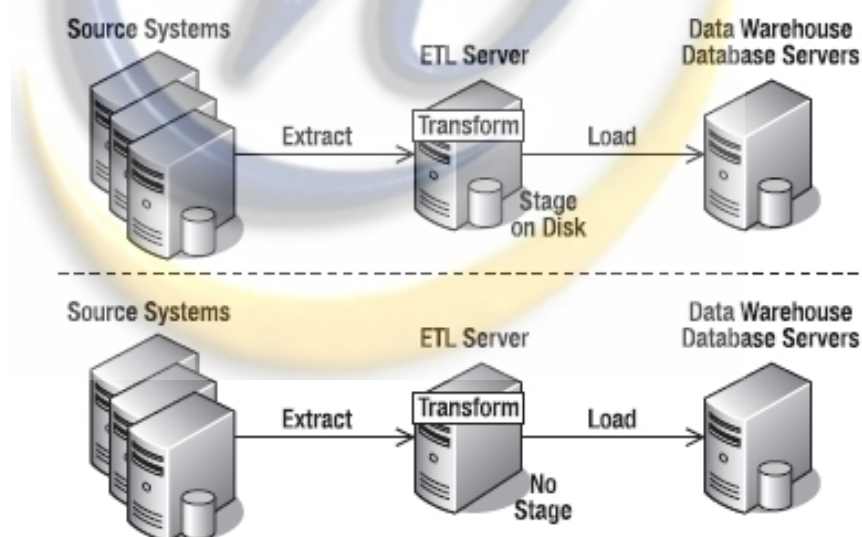
1. *Data Mart* yakni merupakan suatu bagian pada *data warehouse* yang mendukung pembuatan laporan dan analisa data pada suatu unit, bagian atau operasi pada suatu perusahaan.
2. *On-Line Analytical Processing* yakni merupakan suatu pemrosesan basisdata yang menggunakan tabel fakta dan dimensi untuk dapat menampilkan berbagai macam bentuk laporan, analisis, *query* dari data yang berukuran besar.
3. *On-Line Transaction Processing* yakni merupakan suatu pemrosesan yang menyimpan data mengenai kegiatan operasional transaksi sehari-hari.
4. *Dimension Table* yakni merupakan tabel yang berisikan kategori dengan ringkasan data detail yang dapat dilaporkan. Seperti laporan laba pada tabel fakta dapat dilaporkan sebagai dimensi waktu yang berupa perbulan, perkwartal dan pertahun.
5. *Fact Table* yakni merupakan tabel yang umumnya mengandung angka dan data history dimana *key* (kunci) yang dihasilkan sangat unik, karena *key* tersebut terdiri dari *foreign key* (kunci asing) yang merupakan *primary key* (kunci utama) dari beberapa dimensi tabel yang berhubungan.

6. *Decision Support System* yakni merupakan sistem yang menyediakan informasi kepada pengguna yang menjelaskan bagaimana sistem ini dapat menganalisa situasi dan mendukung suatu keputusan yang baik [3].

2.4.2 Data Staging dan ETL

Merupakan fase yang terjadi ketika mengintegrasikan data ke dalam data *warehouse*. Tiga fungsi utama yang perlu dilakukan untuk membuat data siap digunakan pada data *warehouse* adalah *Extraction*, *Transformation* dan *Loading*. *Transformation* merupakan proses yang mempunyai peran dalam melakukan perubahan dan integrasi skema serta struktur yang berbeda-beda ke dalam skema dan struktur yang terdefinisi dalam data *warehouse*.

Ketiga fungsi ini terdapat pada *staging area*. Gambar 3 menjelaskan mengenai *data staging* dan proses ETL :



Gambar 2.1 : Data Staging dan ETL dalam Data warehouse (Paul Lane,2002)

Sumber : <http://blog.umy.ac.id/fajarrianda/files/2014/10/1.png>

1. *Extraction*

Bagian pertama dari suatu proses ETL adalah meng-ekstrak data dari sumber data. Disebut ekstrak, Karena proses mengubah data ke dalam suatu format yang berguna untuk proses

transformasi. Pengambilan data ini tidak mengambil keseluruhan data yang ada di *database* operasional, melainkan hanya mengambil data matang saja. Pada proses ekstraksi data terdapat proses *cleansing* data dimana data operasional yang telah diambil atau dibaca akan diperbaiki dari kesalahan-kesalahan pada input data, redundansi data, nilai *field* yang tidak konsisten, dan sebagainya. Proses ini meliputi penyaringan data yang digunakan langsung dalam *data warehouse*, dapat langsung dimasukkan langsung dalam *data warehouse* atau dimasukkan dalam tempat penampungan sementara terlebih dahulu. Pada hakikatnya bagian dari ekstraksi melibatkan penguraian dari data yang telah diekstrak, menghasilkan suatu pengecekan jika data bertemu dengan suatu struktur atau pola yang diharapkan. Jika bukan, data tersebut mungkin ditolak secara keseluruhan.

2. *Transformation*

Proses yang ke dua adalah transformasi data yang telah diekstrak ke dalam format yang diperlukan. Hal ini perlu dilakukan mengingat data yang diambil berasal dari sumber yang berbeda yang kemungkinan memiliki standarisasi ke dalam format umum yang disepakati dan digunakan dalam *data warehouse*. Aturan-aturan data *transformation* antara lain :

1. *Extracting*

Mengambil data dari sumber operasional dalam “as is” status, dan sumber-sumber data berasal dari *mainframes* yang hampir seluruhnya berupa *database relasional*.

2. *Conditioning*

Perubahan tipe data dari sumber data ke target data (*data warehouse*) tetapi tetap dalam bentuk *database relasional* misal: dilakukan perubahan tipe data atau penamaan data *field* yang ada di *database relasional* ke

dalam data *warehouse* dengan format yang dimengerti dan disesuaikan dengan kebutuhan.

3. *Scrubbing*

Digunakan untuk membersihkan/meningkatkan kualitas data.

4. *Merging*

Proses penggabungan data dengan memperhatikan aturan-aturan dalam mengintegrasikan data.

5. *House Holding*

Mengidentifikasi semua anggota dari *household* (berada di alamat yang sama) untuk mengirimkan *report*. Menjamin hanya satu jenis dokumen saja yang dikirimkan kepada setiap *household* karena mengirimkan *multiple report* akan mengakibatkan naiknya “*cost*” pada *data warehouse*.

6. *Enrichment*

Mengambil data dari sumber eksternal (diluar *data source* yang seharusnya) untuk memperkaya data operasional yang sudah ada.

7. *Scoring*

Komputasi probabilitas tentang sebuah *event*, sebagai contoh menghitung kemungkinan pelanggan membeli produk baru.

8. *Delta Updating*

Proses *update* yang hanya meng-*update* data baru saja tanpa menghasilkan duplikasi *record*.

9. *Refresh*

Proses yang mempropagasi terjadinya proses peng-*updatean* dari sumber data ke *data warehouse*, caranya misalnya melakukan *refresh* secara periodik (setiap malam/minggu) atau setelah *event-event* yang signifikan,

ketika *data warehouse* membutuhkan data yang bersangkutan, atau berdasarkan kebijakan *refresh* yang dibuat oleh *administrator* berdasarkan kebutuhan pengguna.

10. *Validating*

Proses pemeriksaan kebenaran data yang dihasilkan dari penggabungan.

3. *Loading*

Fase *loading* merupakan tahapan yang berfungsi untuk memasukkan data ke dalam target akhir, yang biasanya ke dalam suatu *data warehouse*. *Data loading* adalah memindahkan data ke *data warehouse*.

2.4.3 Perancangan Data Warehouse

Salah satu dari metodologi perancang *data warehouse* adalah *Nine-Step Methodology* oleh Kimball (1996) yang memiliki 9 (sembilan) langkah sebagai berikut:

1. Memilih Proses (*Choosing The Process*)

Proses (fungsi) mengacu pada subjek masalah dari *data mart* tertentu. *Data mart* yang pertama kali dibangun haruslah tepat waktu, sesuai dengan anggaran, dan dapat menjawab pertanyaan-pertanyaan bisnis yang penting.

2. Memilih Sumber (*Choosing The Grain*)

Memilih *grain* berarti memutuskan secara pasti apa yang dinyatakan oleh *record* dari tabel fakta. Hanya dengan telah terpilihnya *grain* untuk tabel fakta maka kita dapat mengidentifikasi dimensi. Keputusan *grain* untuk tabel fakta juga menentukan *grain* untuk setiap dimensi pada tabel fakta.

3. Mengidentifikasi dan Penyesuaian Dimensi (*Identifying and Conforming The Dimensions*)

Dimensi menentukan konteks untuk memberikan pertanyaan mengenai fakta-fakta dalam tabel fakta. Kumpulan dimensi yang dibangun dengan baik membuat *data mart* dapat dimengerti dan mudah untuk digunakan. Dimensi diidentifikasi dalam detail yang cukup untuk mendeskripsikan data seperti *client* dan *properties* dari *grain* yang tepat. Jika ada dimensi yang muncul dalam dua *data mart*, maka kedua *data mart* tersebut harus memiliki dimensi yang sama, atau salah satu *data mart* adalah subset matematis dari *data mart* yang lain. Hanya dengan cara ini dua *data marts* berbagi satu atau lebih dimensi pada aplikasi yang sama. Ketika sebuah dimensi digunakan oleh lebih dari satu *data mart* dan dimensi tersebut tidak disinkronisasikan antar *data mart* maka seluruh *data warehouse* akan gagal, karena dua *data mart* tidak dapat digunakan pada saat yang bersamaan.

4. Memilih Fakta (Choosing The Fact)

Grain dari tabel fakta menentukan fakta mana yang dapat digunakan pada *data mart*. Semua fakta harus dinyatakan berdasarkan tingkatan yang tersirat oleh *grain*. Fakta tambahan dapat ditambahkan ke dalam tabel fakta pada setiap waktu dengan catatan fakta tersebut konsisten dengan *grain* dari tabel.

5. Menyimpan Perhitungan Awal dalam Tabel Fakta (*Storing Pre-Calculation in The Fact Table*)

Setelah fakta dipilih, setiap fakta harus dikaji ulang untuk menentukan apakah ada kemungkinan untuk melakukan *pre-calculations*. *Pre-calculations* terjadi ketika fakta terdiri dari *statement* untung dan rugi.

6. Melihat Kembali Tabel Dimensi (*Rounding Out The Dimension Tables*)

Pada langkah ini, kita kembali mengkaji tabel dimensi dan menambahkan sebanyak mungkin deskripsi teks ke dimensi. Teks deskripsi haruslah seintuitif mungkin dan dapat dimengerti oleh

pengguna. Kegunaan dari *data mart* ditentukan oleh cakupan dan sifat atribut pada tabel dimensi.

7. Memilih Durasi *Database* (*Choosing The Duration of Database*)

Durasi mengukur berapa lama tabel fakta dapat disimpan. Pada banyak perusahaan, ada ketentuan untuk melihat pada periode waktu yang sama satu atau dua tahun sebelumnya. Tabel fakta yang sangat besar akan mengakibatkan setidaknya dua masalah yang signifikan pada *data warehouse*. Pertama, bertambahnya kesulitan untuk menjadikan data lama yang semakin bertambah sebagai sumber. Semakin lama suatu data, semakin banyak masalah dalam membaca dan menginterpretasikan *file* lama tersebut. Kedua, kebutuhan dimensi menggunakan versi yang lama, bukan versi yang baru.

8. Menelusuri Perubahan dari Dimensi secara Perlahan (*Tracking Slowly Changing Dimension*)

Pada tahap ini, *data warehouse* memperhatikan proses dimensi yang semakin tua seiring dengan berjalannya waktu. Untuk itu perlu dilakukan *update* agar *data warehouse* selalu konsisten.

Terdapat tiga tipe dasar perubahan dimensi secara perlahan:

- a. Tipe 1: Perubahan data secara langsung atau *update table* dimensi.
 - b. Tipe 2: Perubahan data membentuk *record* baru dengan *surrogate key* yang berbeda.
 - c. Tipe 3: Perubahan data akan membentuk atribut atau kolom baru pada tabel dimensi.
9. Memutuskan Prioritas *Query* dan Tipe *Query* (*Deciding The Query Priorities and The Query Models*)

Pada tahap ini dipertimbangkan masalah perancangan fisik (*physical design*). Masalah utama pada perancangan fisik yang mempengaruhi persepsi pengguna akhir dari *data mart* adalah urutan penyusunan tabel fakta pada *disk* dan adanya *pre-stored summaries* dan agregasi [4].

2.4.4 Pemodelan Dimensional

Menurut Connolly dan Begg (2005, p1183), pemodelan dimensional adalah sebuah teknik perancangan logical yang bertujuan untuk mempresentasikan data ke dalam sebuah standar, bentuk intuitif yang dapat diakses dengan performa yang tinggi. Setiap model dimensional terdiri dari sebuah tabel dengan sebuah *primary key* komposit yang disebut dengan tabel fakta, dan sekumpulan tabel yang lebih kecil yang disebut dengan tabel dimensi. Setiap tabel dimensi memiliki sebuah *primary key* (nonkomposit) sederhana yang berkorespondensi tepat dengan satu key komposit pada tabel fakta. Dengan kata lain, *primary key* dari tabel fakta terbuat dari dua atau lebih *foreign key*. Karakteristik dengan struktur yang seperti bintang ini disebut dengan *star schema* atau *star join* [5].

1. *Star Schema*

Star schema merupakan relasi dari beberapa tabel dimensi yang terpusat pada tabel fakta. Sebuah skema dikatakan *star* jika tabel dimensinya melakukan *join* secara langsung ke tabel fakta. Pemodelan dengan *star schema* ini membutuhkan ruang yang lebih untuk analisa multidimensi dalam *data warehouse*. *Star schema* ini mengoptimalkan performa dalam melakukan *query* dan memberikan kecepatan *respon time* karena informasi dari masing masing level disimpan dalam tiap barisnya [J.Han & M. Kamber, 2006 : 114].

2. *Snowflake Schema*

Snowflake schema merupakan perluasan dari *star schema* dimana satu atau lebih dimensinya diperoleh dari beberapa tabel. Dalam skema ini, hanya satu dimensi utama yang dihubungkan dengan tabel fakta. Sedangkan dimensi-dimensi lainnya dihubungkan dengan tabel dimensi utama. Skema *snowflake* membutuhkan biaya yang besar untuk melakukan proses *query*-nya karena beberapa tabel terhubung secara rumit sehingga akan mengakibatkan proses pencarian data dalam *data warehouse* menjadi lambat [J.Han & M. Kamber, 2006 : 128].

2.5 OLAP (*Online Analytical Processing*)

OLAP singkatan dari *On-Line Analytical Processing*. Secara mendasar OLAP adalah suatu metode khusus untuk melakukan analisis data yang terdapat di dalam media penyimpanan data (*database*) dan kemudian membuat laporannya sesuai dengan permintaan *user* [Yudhi Hermawan, 2005 : 110]. OLAP adalah bagian dari kategori yang lebih global dari pemikiran bisnis, yang juga merangkum hubungan antara pelaporan dan penggalan data. Aplikasi khusus dari OLAP adalah pelaporan bisnis untuk penjualan, pemasaran, manajemen pelaporan, manajemen proses bisnis (MPB), penganggaran dan peramalan, laporan keuangan dan bidang-bidang yang serupa. Istilah OLAP merupakan perampingan dari istilah lama database OLTP (*Online Transaction Processing*). Database yang dikonfigurasi untuk pelayanan OLAP model data multidimensi, bisa digunakan untuk analisis kompleks dan kueri khusus (*ad hoc*) dengan suatu laju waktu eksekusi. Mereka meminjam aspek *database* navigasi dan *database* hierarki yang lebih cepat daripada yang sefamiliinya.

OLAP akan menampilkan data dalam sebuah tabel yang dinamis, yang secara otomatis akan meringkas data ke dalam beberapa irisan data yang berbeda dan mengizinkan *user* untuk secara interaktif melakukan perhitungan serta membuat laporan. *Tool* untuk membuat laporan tersebut adalah tabel itu sendiri yaitu dengan melakukan *drag* terhadap kolom dan baris. *User* dapat mengubah bentuk laporan dan menggolongkannya sesuai dengan keinginan dan kebutuhan *user* dan OLAP *engine* secara otomatis akan mengkalkulasi data tersebut [Yudhi Hermawan, 2005 : 105]. Berikut adalah operasi-operasi yang terdapat pada OLAP yang merupakan keunggulan dalam proses analisa OLAP [J. Han & M. Kamber, 2006 : 123] :

a. *Roll-up* dan *Drill Down*

Operasi ini merupakan proses agregasi data. Proses *drill down* adalah proses penampilan data dalam bentuk yang lebih detail. Proses pendetailan ini berdasarkan konsep hirarki data yang telah terformat sebelumnya. Kebalikannya yaitu *consolidation* yaitu penggabungan atau penyatuan data ke dalam level yang lebih tinggi.

b. *Slice dan Dice*

Operasi *slicing* dan *dicing* merupakan kemampuan OLAP untuk melakukan pemilihan subset pada suatu data. Proses *slicing* adalah proses pemotongan data pada *cube* berdasarkan nilai pada satu atau beberapa dimensi. Sedangkan *dicing* adalah pemotongan hasil *slicing* menjadi bagian subset data yang lebih kecil.

c. *Pivoting*

Pivoting adalah kemampuan OLAP untuk mengubah berbagai sudut pandang data. Dengan operasi ini perubahan perspektif sudut pandang menjadi lebih mudah. Rotasi dapat dilakukan dengan memutar masing-masing sumbu dari *cube* yang dikehendaki untuk menampilkan data dari berbagai sudut pandang.[1]

2.5.1 Teknik OLAP (*Online Analytical Processing*)

Teknik OLAP dapat dirangkum menjadi 5 garis besar yaitu *Fast Analysis of Shared Multidimensional Information* atau disingkat menjadi FASMI yang masing-masing berarti sebagai berikut :

- a. *Fast*, berarti sistem ditargetkan untuk memberikan respon terhadap user dengan secepat mungkin, sesuai dengan analisis yang dilakukan.
- b. *Analysis*, berarti sistem dapat mengatasi berbagai logika bisnis dan analisis statistik yang relevan dengan aplikasi dan user, dan mudah.
- c. *Shared*, berarti sistem melaksanakan seluruh kebutuhan pengamanan data, jika dibutuhkan banyak akses penulisan terhadap data, disesuaikan dengan level dari user. Tidak semua aplikasi membutuhkan user untuk menulis data kembali. Sistem harus dapat meng-*handle multiple update* dalam satu waktu secara aman.
- d. *Multidimensional*, berarti sistem harus menghasilkan *conceptual view* dari data secara multidimensional, meliputi *full support* untuk hierarki dan multiple hierarki. Hal ini merupakan cara yang *logic* untuk menganalisis bisnis dan organisasi.
- e. *Information*, adalah semua data dan informasi yang dibutuhkan dan relevan untuk aplikasi. Kapasitas produk OLAP berbeda untuk

menghandle input data tergantung beberapa pertimbangan meliputi duplikasi data, RAM yang dibutuhkan, penggunaan *disk space*, *performance*, integrasi dengan data *warehouse*, dan lainnya.

2.5.2 Karakteristik OLAP (*Online Analytical Processing*)

Adapun karakteristik dari OLAP, yaitu:

- a. Mengizinkan *user* melihat data dari sudut pandang *logical* dan *multidimensional* pada *data warehouse*.
- b. Memfasilitasi *query* yang kompleks dan analisa bagi *user*.
- c. Mengizinkan user melakukan *drill down* untuk menampilkan data pada level yang lebih detil atau *roll up* untuk agregasi dari satu dimensi atau beberapa dimensi.
- d. Menyediakan proses kalkulasi dan perbandingan data.
- e. Menampilkan hasil dalam bentuk number termasuk dalam tabel dan grafik.

2.5.3 Kelebihan Menggunakan OLAP (*Online Analytical Processing*)

- a. Dapat meningkatnya produktivitas bisnis, *IT developers*, dan seluruh organisasi.
- b. Akses yang lebih terkendali terhadap informasi yang dapat meningkatkan efektivitas pengambilan keputusan.
- c. Mempercepat respon terhadap permintaan pasar.
- d. Mengurangi “*backlog*” pengembangan aplikasi bagi staf IT dengan membua tpemakai akhir dapat merubah schema dan membangun model sendiri.
- e. Penyimpanan pengawasan organisasi melalui integritas data korporasi sebagai aplikasi OLAP tergantung pada data *warehouse* dan sistem OLTP untuk memperbaharui sumber tingkatan data mereka.
- f. Mengurangi aktifitas *query* dan lalu lintas jaringan pada sistem OLTP atau pada data *warehouse*.
- g. Meningkatkan hasil dan keuntungan secara potensial dengan mengizinkan organisasi untuk merespon permintaan pasar lebih cepat.

2.5.4 Manfaat OLAP (*Online Analytical Processing*)

Kesuksesan implementasi aplikasi OLAP meningkatkan produktivitas manajer bisnis, pengembang, dan organisasi secara keseluruhan. Fleksibilitas yang melekat dari sistem OLAP berarti pengguna bisnis aplikasi OLAP dapat menjadi lebih mandiri. OLAP memungkinkan manajer untuk memodelkan masalah yang tidak mungkin dilakukan dengan menggunakan sistem yang kurang fleksibel dengan waktu tanggapan yang panjang dan tidak konsisten.

Pengembang Teknologi Informasi juga mendapat keuntungan dari penggunaan perangkat lunak OLAP dengan benar. Dengan menggunakan perangkat lunak yang khusus dirancang untuk OLAP, pengembang dapat mengirimkan aplikasi untuk pengguna bisnis lebih cepat, menyediakan layanan yang lebih baik. Pengiriman cepat aplikasi juga mengurangi backlog aplikasi.

Pemanfaatan OLAP dengan pola analisis seperti berikut ini :

- a. Meringkas dan mengumpulkan sejumlah besar data.
- b. Melakukan *filtering*, pengurutan, dan memberikan peringkat (rangking).
- c. Membandingkan beberapa set dari data.
- d. Membuat sketsa/bagan/diagram.
- e. Menganalisis dan menemukan pola dari data.
- f. Menganalisis kecenderungan data.

2.5.5 Perbedaan OLTP (*Online Transaction Processing*) dan OLAP (*Online Analytical Processing*)

1. *User*

Dalam OLTP, penggunaannya adalah IT Profesional sedangkan OLAP penggunaannya adalah *Knowledge worker* maksudnya penggunaannya adalah seorang yang bertindak dalam *subyek* tertentu, atau petinggi dalam suatu perusahaan.

2. *Function*

OLTP digunakan sehari-hari untuk proses bisnis seperti toko atau swalayan, sedangkan OLAP digunakan untuk pengambilan keputusan.

3. **Design Database**

Desain dalam OLTP bersifat *Entity Relational* atau databasenya dinormalisasi dulu sebelum digunakan. Untuk OLAP desain databasenya di de-normalisasi.

4. **Data**

Dalam OLTP datanya adalah hari ini, *update* setiap saat sedangkan OLAP datanya adalah sekarang dan hari ini yang berguna untuk melakukan analisis ke depan.

5. **Penggunaan**

OLTP digunakan setiap saat, sedangkan OLAP digunakan seperlunya saja.

6. **Access**

OLTP aksesnya bias ditulis, dibaca dan lain-lain. Sedangkan OLAP sering dibaca karena digunakan untuk analisa.

7. **Unit Pekerjaan**

Kalau OLTP pekerjaannya hanya sederhana misalnya transaksi dalam swalayan. Untuk OLAP query untuk menampilkan data sangat kompleks

8. **Jumlah rekaman yang di akses**

Kalau OLTP sekitar ratusan sampai ribuan, tapi jika OLAP data yang diakses bisa sampai jutaan bahkan milyaran.

9. **Jumlah Pengguna**

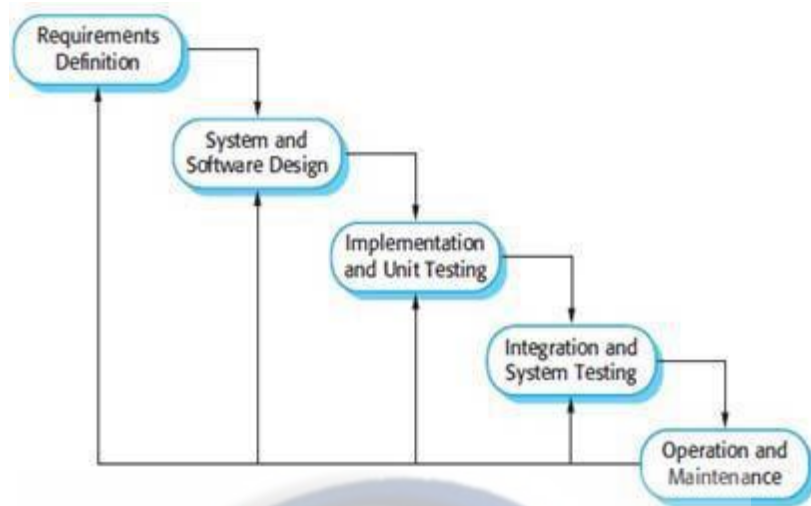
Untuk OLTP penggunanya adalah puluhan, tapi kalau OLAP penggunanya bisa sampai ratusan bahkan ribuan

10. **Ukuran Database**

Ukuran database untuk OLTP sekitar MB-GB, sedangkan OLAP bisa sampai GB-TB

2.6 **Waterfall Model**

Menurut (I Sommerville, 2011) terdapat 5 tahapan pada *waterfall* model, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and sistem testing*, dan *operation and maintenance*.



Gambar 2.2 Metode Waterfall (I Sommerville, 2011)

Sumber : <https://rpl07.files.wordpress.com/2007/06/28.gif>

1. Requirement Analysis and Definition

Merupakan tahapan penetapan fitur, kendala, serta tujuan sistem melalui proses wawancara yang dilakukan dengan pengguna sistem. Proses ini bertujuan untuk menentukan spesifikasi sistem yang akan dibangun.

2. Sytem and Software Design

Tahapan ini merupakan tahapan pembentukan suatu arsitektur sistem berdasarkan yang telah ditentukan pada proses sebelumnya. Tahapan ini berfungsi sebagai gambaran abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.

3. Implementation and Sistem Testing

Pada tahapan ini dilakukan proses implementasi desain yang telah dibuat pada tahapan sebelumnya kedalam bentuk program dengan menggunakan bahasa pemrograman. Setiap unit akan dilakukan pengujian untuk memastikan semua fungsi dengan yang diharapkan.

4. Integration and Sistem Testing

Pada tahapan ini setiap unit program akan dilakukan pengintegrasian antara satu dengan yang lainnya, serta dilakukan proses ujicoba sebagai satu sistem yang utuh untuk memastikan sistem yang dibangun telah memenuhi persyaratan yang ada.

5. *Operation and Maintenance*

Tahapan ini merupakan tahapan terakhir dimana pada tahapan ini dilakukan proses penggunaan sistem. Pada tahapan ini dapat dilakukan perbaikan apabila ada error ditemukan.

2.7 *Query*

Query adalah kemampuan untuk menampilkan suatu data dari database dimana data diambil dari tabel-tabel yang ada di dalam database sesuai dengan kebutuhan. Bahasa *query* merupakan bahasa khusus yang digunakan untuk melakukan query pada basis data. Contoh penggunaan bahasa query adalah `SELECT * FROM KATEGORI WHERE KD_WAKTU=2`. Query tersebut meminta untuk menampilkan semua record yang terdapat pada tabel kategori dengan `KD_WAKTU=2`.

2.8 *Pentaho*

Pentaho adalah sebuah nama perusahaan dan juga nama dari produk yang dimiliki oleh perusahaan tersebut. *Pentaho* merupakan sebuah perusahaan yang khusus membuat *software* untuk keperluan *data warehouse* (DWH) dan *business intelligence* (BI). Produk-produk yang ditawarkan dalam dua bentuk edisi, yaitu *Enterprise Edition* (EE) dan *Community Edition* (CE).

Perbedaan produk EE dan CE terletak pada :

1. Produk EE memiliki fitur-fitur yang tidak disediakan para produk CE.
2. Untuk menggunakan produk EE diharuskan membayar biaya berlangganan tahunan dan akan mendapatkan dukungan layanan langsung dari *pentaho corporation*.
3. Produk CE bersifat gratis selama digunakan sesuai dengan perjanjian lisensi yang telah disediakan, produk CE tidak mendapatkan dukungan dari *pentaho corporation*.

2.9 *Pentaho Data Integration (PDI)*

Pada tahun 2001, Matt Casters membuat sebuah perangkat ETL yang diberi nama *kettle* yang merupakan singkatan dari *KDE ETL Environment*, karena pada awalnya perangkat ini ditujukan untuk dapat dijalankan di atas *K Desktop*

Environmernt (KDE) linux. ETL pada *kettle* merupakan singkatan dari *Extraction, Transpormation, Transformation*, dan *Loading*.

Karena *kettle* memiliki beragam keunggulan disbanding perangkat ETL *open source* lainnya, pada april 2006, kode sumber *kettle* resmi diambil oleh *pentaho*. Oleh *pentaho*, *kettle* diberi nama *pentaho data integration* (PDI).

2.10 Metode *Moving Average*

Moving Average menyediakan metode sederhana untuk pemulusan data masa lalu, metode ini berguna untuk peramalan ketika tidak terjadi tren. Tujuan utama dari penggunaan rata-rata bergerak adalah untuk menghilangkan atau mengurangi acakan dalam deret waktu. Dengan peramalan *moving average* dilakukan dengan mengambil sekelompok nilai pengamatan, mencari rata-ratanya, lalu menggunakan rata-rata tersebut sebagai ramalan untuk periode berikutnya. Istilah rata-rata bergerak digunakan karena setiap kali ada data observasi baru tersedia, maka angka rata-rata yang baru dihitung dan dipergunakan sebagai ramalan. Rumus dari perhitungan *moving average* adalah sebagai berikut :

$$\hat{f}_t = \frac{f_{t-1} + f_{t-2} + f_{t-3} + \dots + f_{t-n}}{n}$$

Dimana :

n = Jumlah periode yang digunakan sebagai dasar peramalan

\hat{f}_t = Ramalan permintaan (*real*) untuk periode t

f_t = Permintaan aktual pada periode t

Pada penelitian ini penulis dalam melakukan prediksi penjualan menggunakan jenis metode *single moving average* :

2.10.1 *Single Moving Average*

Menentukan ramalan dengan metode *single moving average* cukup mudah dilakukan. Bila akan menerapkan 12 bulan rata-rata bergerak maka ramalan pada bulan Januari dihitung sebesar rata-rata dari 12 bulan sebelumnya, yaitu bulan Januari, Februari, Maret, April, Mei, Juni Juli,

Agustus, September, Oktober, November, dan Desember. Persamaan matematis dari teknik ini adalah sebagai berikut :

$$F_{t+1} = \frac{X_t + X_{t-1} + X_{t-2} + \dots + X_{t-n+1}}{n}$$

Dimana :

F_{t+1} = Ramalan untuk periode ke $t + 1$

X_t = Nilai riil periode ke t

n = Jangka waktu rata-rata bergerak

Metode *single moving average* menurut Subagyo (1986 : 60) mempunyai sifat khusus.

1. Untuk menentukan ramalan pada periode yang akan datang memerlukan data historis selama jangka waktu tertentu.
2. Semakin panjang jangka waktu *moving average*, akan menghasilkan *moving average* yang semakin halus. Artinya pada *moving averages* yang jangka waktunya lebih panjang, perbedaan ramalan terkecil dengan ramalan terbesar menjadi kecil. metode Rata-rata bergerak orde satu (*single moving average*) ini biasanya lebih cocok digunakan untuk melakukan *forecast* hal-hal yang bersifat *random*, artinya tidak ada gejala trend naik maupun turun, musiman, dan sebagainya, melainkan sulit diketahui polanya

Metode ini mudah menghitungnya dan sederhana, tetapi mempunyai kelemahan-kelemahan sebagai berikut;

1. Perlu data historis yang cukup,
2. Data tiap periode diberi *weight* (bobot) sama
3. Kalau fluktuasi data tidak random tidak menghasilkan peramalan (*forecasting*) yang baik. (Subagyo,1986: 11).

Menghitung tingkat kesalahan pada metode *single moving average* :

1. *Mean Absolute Error* (MAE)

Mean Absolute Error atau nilai tengah kesalahan absolut adalah rata-rata mutlak dari kesalahan meramal, tanpa menghiraukan tanda positif maupun negatif.

$$MAE = \frac{\sum |Y_t - F_t|}{n}$$

Keterangan :

Y_t = Data aktual pada periode t

F_t = Peramalan periode t

2. *Mean Squared Error* (MSE)

MSE merupakan metode alternatif untuk mengevaluasi teknik peramalan masing-masing kesalahan (selisih data aktual terhadap data peramalan) dikuadratkan, kemudian dijumlahkan dan dibagi dengan jumlah data. MSE dihitung dengan rumus :

$$MSE = \frac{\sum (Y_t - F_t)^2}{n}$$

Keterangan :

Y_t = Data aktual pada periode t

F_t = Peramalan periode t [9]