BAB II

TINJAUAN PUSTAKA

Teori merupakan dasar yang digunakan dalam penyusunan aplikasi data *warehouse* yang akan dibangun. Pada bab ini akan dijelaskan mengenai tinjauan pustaka dan teori yang berkaitan dengan data Warehosue pada PT. KACA PATRI

2.1 Ruang Lingkup Perusahaan

2.1.1 Sejarah Singkat Perusahaan

PT. Kaca Patri merupakan perusahaan yang bergerak dibidang industry penjualan dan pembelian berbagai jenis kaca, dimana perusahaan ini sedang berkembang dan telah memiliki beberapa cabang. PT. Kaca Patri ini didirikan oleh bapak P.Naibaho pada tanggal 21 juni 2005 hingga saaat ini. perusahaan ini merupakan suatu perusahaan yang sudah berkembang, akan tetapi masih tetap membutuhkan perubahan dan pembaharuan yang dapat mempertahankan eksistensinya, karena perusahaan ini memiliki pelanggan dari perorangan maupun dari perusahaan lain yang melakukan pemesanan barang dan membutuhkan jenis kaca secara cepat dan sesuai dengan apa yang mereka butuhkan.

PT. Kaca Patri ini memiliki komitmen kualitas dan hasil kerja dalam memberikan suatu pelayanan. Hal ini mendorong PT. Kaca Patri untuk bekerja secara disiplin dan memiliki loyalitas kerja yang mantap serta berdedikasi tinggi untuk

menciptakan profesionalisme dan perusahaan yang bonafit. Oleh karena itu kualifikasi personal dan profesional perusahaan dalam kancah persaingan bisnis yang semakin tajam, sehingga mampu mengantisipasi permasalahan yang berkaitan dengan bisnis yang digeluti.

2.1.2 Visi dan Misi

Visi adalah cara pandang jauh ke depan kemana instansi harus dibawa agar dapat eksis, antisipatif dan inovatif, merupakan gambaran ke depan yang menantang tentang keadaan masa depan yang di inginkan oleh instansi. Visi menggambarkan aspirasi masa depan tanpa menspesifikasi cara-cara untuk mencapainya, visi yang efektif adalah visi yang mampu membangkitkan inspirasi.Sedangkan misi adalah sesuatu yang harus dilaksanakan oleh instansi agar tujuan organisasi dapat terlaksana dan berhasil dengan baik. Misi merupakan bentuk yang didambakan di masa depan (*what do they want to be)*, sebuah pernyataan yang menegaskan visi lewat pilihan bentuk atau garis besar jalan yang akan diambil untuk sampai pada visi yang telah lebih dulu dirumuskan. Keduanya tidak memiliki dimensi ukur *kuantitatif* (persentase, besaran waktu, dll). Sebagai konsep yang ideal visi-misi ini harus diterjemahkan lagi dalam konsep yang lebih nyata dan terukur yaitu tujuan (objective) yaitu tujuan yang nyata (*real*).

Adapun visi pada Pusat Material Seni Bangunan ini adalah:

“Menjadi perusahaan yang senantiasa mampu bersaing dan tumbuh berkembang dengan sehat serta terpercaya dalam menjalankan usaha dibidangnya”.

Misi dari PT.Kaca Patri adalah:

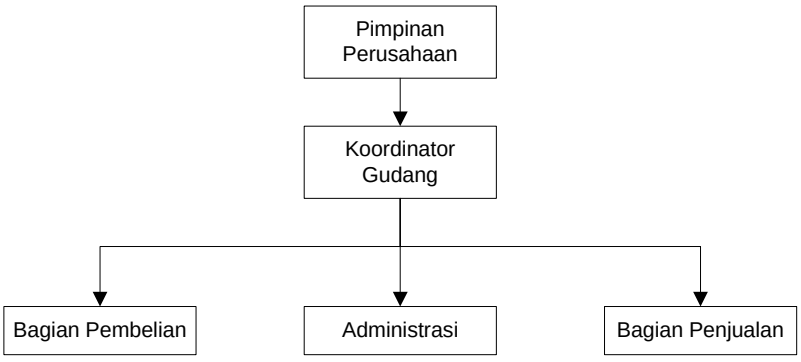
1. Mengelola perusahaan secara profesional, terbuka dan mematuhi peraturan perundangan yang berlaku untuk menghasilkan kinerja yang ekselen.
2. Meningkatkan nilai tambah dan kinerja finansial perusahaan dengan memperhatikan efisiensi biaya dan efektifitas program.
3. Menghasilkan laba yang pantas untuk mendukung pengembangan perusahaan.
4. Memproduksi berbagai jenis kaca yang terkait dengan kebutuhan industri dan masyarakat dengan mutu, harga dan pasokan yang berdaya saing tinggi melalui pengelolaan yang profesional demi kepuasan pelanggan.
5. Memberikan penghargaan kepada para pegawai melalui pemberian kesejahteraan yang memadai, penyediaan lingkungan kerja yang aman, sehat dan nyaman.
6. Menjalin kemitraan kerja sama dengan pemasok dan penyalur yang saling menguntungkan.

2.1.3 Struktur Perusahaan

Struktur organisasi merupakan gambaran secara grafik yang terbentuk struktur kerja dari suatu struktur organisasi. Struktur organisasi hanya dapat menunjukan hubungan wewenang yang formal dan tidak dapat menggambarkan seberapa besar wewenang, tanggung jawab dan deskripsi pekerjaan yang terperinci.

Struktur organisasi ini cukup penting karena merupakan bentuk kegiatan dan hubungan antara berbagai sub unit bagian-bagian didalam perusahaan dengan mengetahui struktur organisasi dapat diperoleh satu gambaran tentang bagian-bagian yang ada didalam suatu organisasi. Struktur organisasi pada PT. Kaca Patri ini merupakan struktur oraganisasi yang kuat dan mapan, sehingga sangat efektif dan efisien dalam pelaksaan kegiatan untuk mecapai tujuan perusahaan.

Berikut adalah gambaran struktur organisasi yang lebih rinci dari PT.Kaca Patri



Gambar 2.1 Struktu Organisasi PT. Kaca Patri

Tugas masing – masing pada struktur organisasi PT. Kaca Patri

1. Tugas pimpinan perusahaan adalah memantau dan menilai keseluruhan dari kemajuan perusahaan serta bertanggung jawab untuk pengambilan keputusan bagi perusahaan.
2. Tugas koordinator gudang adalah sebagai penanggung jawab dari semua transaksi yang terjadi di gudang perusahaan.
3. Tugas bagian pembelian adalah mangatur jadwal pembelian barang dan melakukan pemesanan barang untuk persediaan gudang serta mencatat semua transaksi pembelian.
4. Tugas administrasi adalah mencatat dan menyusun laporan-laporan keuangan dari perusahaan.
5. Tugas bagian penjualan adalah mengatur jadwal penjualan dan melakukan penjualan untuk memenuhi permintaan pelanggan serta mencatat semua transaksi penjualan.

2.2 Landasan Teori

2.2.1 Data

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berujut suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupunsuatu

konsep. Sedangkan menurut Wawan dan Munir (2006:6) Selain deskripsi dari sebuah fakta, data dapat pula merepresentasikan suatu objek atau kejadian (*event*) “

Data juga bisa menggambarkan sebuah representasi fakta yang tersusun secara terstruktur, dengan kata lain bahwa *“Generally, data represent a structured codification of single primary entities, as well as of transactions involving two or more primary entities* .” (Vercellis, 2009: 6). Sedangkan menurut Wawan dan Munir (2006:6) Selain deskripsi dari sebuah fakta, data dapat pula merepresentasikan suatu objek atau kejadian (*event*) “

Dengan demikian dapat dijelaskan kembali bahwa data merupakan suatu objek, kejadian, atau fakta yang terdokumentasikan dengan memiliki kodifikasi terstruktur untuk suatu atau beberapa entitas.

2.2.1.1 Jenis Data Menurut Cara Memperolehnya

Jenis data dapat diuraikan berdasarkan cara memperolehnya, yakni :

1. Data Primer

   Data primer adalah secara langsung diambil dari objek penelitian oleh peneliti perorangan maupun organisasi.

   Contoh: Mewawancarai langsung penonton bioskop 21 untuk meneliti preferensi konsumen bioskop.

2. Data Sekunder

   Data sekunder adalah data yang didapat tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi yang dikumpulkan

oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial.

Contoh: Pada peneliti yang menggunakan data statistik hasil riset dari surat kabar atau majalah.

2.2.1.2 Macam–Macam Data Berdasarkan Sumber Data

Maca-macam data dapat dibedakan berdasarkan sumber data yang diperoleh, yakni :

1. Data Internal

   Data internal adalah data yang menggambarkan situasi dan kondisi pada suatu organisasi secara internal.

   Contoh: Data keuangan, data pegawai, data penjualan suatu perusahaan atau organisasi

2. Data Eksternal

   Data eksternal adalah data yang menggambarkan situasi serta kondisi yang ada di luar organisasi.

   Contohnya: Data jumlah penggunaan suatu produk pada konsumen, tingkat preferensi pelanggan, persebaran penduduk, dan lain sebagainya.

2.2.1.3 Klasifikasi Data Berdasarkan Jenis Datanya

Kasifikasi data dapat dibedakan berdasarkan jenis data yang diperoleh, yakni:

1. Data Kuantitatif

   Data kuantitatif adalah data yang dipaparkan dalam bentuk angka-angka.

   Misalnya : jumlah penjualan pada hari raya idul adha.

2. Data Kualitatif

   Data kualitatif adalah data yang disajikan dalam bentuk kata-kata yang yang mengandung makna.

   Contoh: persepsi konsumen terhadap botol air minum dalam kemasan.

2.2.1.4 Pembagian Jenis Data Berdasarkan Sifat Data

Data menurut pembagian jenis data berdasarkan sifat data dapat di bagi sebagai berikut:

1. Data Diskrit

   Data diskrit adalah data yang nilainya adalah bilangan asli.

   Contoh: nilai rupiah dari waktu ke waktu.

2. Data Kontiniu

   Data kontiniu adalah data yang nilainya ada pada suatu interval tertentu atau berada pada nilai yang satu ke nilai yang lainnya.

   Contoh: penggunaan kata sekitar, kurang lebih, kira-kira.

2.2.1.5 Jenis-Jenis Data Menurut Waktu Pengumpulannya

Data juga dapat dibedakan bedakan berdasarkan jenis-jenis data berdasarkan waktu pengumpulannya, yakni:

1. Data Croos Section

   Data cross-section adalah data yang menunjukkan titik waktu tertentu. Contohnya laporan penjualan Pada PT. Kaca Patri Pada tanggal 04-april-2011.

2. Data Times Series atau Berkala

   Data berkala adalah data yang datanya menggambarkan sesuatu dari waktu ke waktu atau periode secara historis.

   Contoh: Data Perkembangan nilai tukar dollar amerika terhadap rupiah dari tahun 2008 sampai 2009.

2.2.2. Data *Warehouse*

2.2.2.1 Pengertian *Data Warehouse*

Pengertian Data *Warehouse* dapat bermacam-macam namun mempunyai inti yang sama, seperti pendapat beberapa ahli berikut ini :

Menurut W.H. Inmon dan Richard D.H., *data warehouse* adalah koleksi data yang mempunyai sifat berorientasi subjek,terintegrasi,time-variant, dan bersifat tetap dari koleksi data dalam mendukung proses pengambilan keputusan management.

Menurut Vidette Poe, *data warehouse* merupakan database yang bersifat analisis dan *read only* yang digunakan sebagai fondasi dari sistem penunjang keputusan.

Menurut Paul Lane, *data warehouse* merupakan database relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, biasanya mengandung *history* data dari proses transaksi dan bisa juga data dari sumber lainnya. *Data warehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung/konsolidasi data dari berbagai macam sumber.

Jadi, *data warehouse* merupakan metode dalam perancangan database, yang menunjang DSS(Decission Support System) dan EIS (Executive Information System). Secara fisik *data warehouse* adalah database, tapi perancangan data *warehouse* dan database sangat berbeda. Dalam perancangan database tradisional menggunakan normalisasi, sedangkan pada *data warehouse* normalisasi bukanlah cara yang terbaik.

Dari definisi-definisi yang dijelaskan tadi, dapat disimpulkan *data warehouse* adalah database yang saling bereaksi yang dapat digunakan untuk *query* dan analisisis, bersifat orientasi subjek, terintegrasi, time-variant,tidak berubah yang digunakan untuk membantu para pengambil keputusan.

Karakteristik data *warehouse* menurut Inmon, yaitu :

1. ***Subject Oriented*** **(Berorientasi subject)**

Data warehouse berorientasi subject artinya data *warehouse* didesain untuk menganalisa data berdasarkan subject-subject tertentu dalam organisasi,bukan pada proses atau fungsi aplikasi tertentu.

Data *warehouse* diorganisasikan disekitar subjek-subjek utama dari perusahaan(customers,products dan sales) dan tidak diorganisasikan pada area-area aplikasi utama(customer invoicing,stock control dan product sales). Hal ini dikarenakan kebutuhan dari data *warehouse* untuk menyimpan data-data yang bersifat sebagai penunjang suatu keputusan, dari pada aplikasi yang berorientasi terhadap data. Jadi dengan kata lain, data yang disimpan adalah berorientasi kepada subjek bukan terhadap proses. Secara garis besar perbedaan antara data operasional dan data *warehouse* yaitu :

Tabel 2.1 Perbedaan Data Operasional dan data *warehouse*

Data Operasional	**Data *Warehouse***
Dirancang berorientasi hanya pada aplikasi dan fungsi tertentu	Dirancang berdasar pada subjek-subjek tertentu(utama)
Focusnya pada desain database dan proses	Focusnya pada pemodelan data dan desain data
Berisi rincian atau detail data	Berisi data-data history yang akan dipakai dalam proses analisis
Relasi antar table berdasar aturan terkini(selalu mengikuti rule(aturan) terbaru)	Banyak aturan bisnis dapat tersaji antara tabel-tabel

2. *Integrated* (Terintegrasi)

Data Warehouse dapat menyimpan data-data yang berasal dari sumber-sumber yang terpisah kedalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Dengan demikian data tidak bisa dipecah-pecah karena data yang ada merupakan suatu kesatuan yang menunjang keseluruhan konsep data *warehouse* itu sendiri. Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara sepeti konsisten dalam penamaan variable,konsisten dalam ukuran variable,konsisten dalam struktur pengkodean dan konsisten dalam atribut fisik dari data.

Contoh pada lingkungan operasional terdapat berbagai macam aplikasi yang mungkin pula dibuat oleh developer yang berbeda. Oleh karena itu, mungkin dalam aplikasi-aplikasi tersebut ada variable yang memiliki maksud yang sama tetapi nama dan format nya berbeda. Variable tersebut harus dikonversi menjadi nama yang sama dan format yang disepakati bersama. Dengan demikian tidak ada lagi kerancuan karena perbedaan nama, format dan lain sebagainya. Barulah data tersebut bisa dikategorikan sebagai data yang terintegrasi karena kekonsistenannya.

3. *Time-variant (*Rentang Waktu*)*

Seluruh data pada data *warehouse* dapat dikatakan akurat atau valid pada rentang waktu tertentu. Untuk melihat interval waktu yang digunakan

dalam mengukur keakuratan suatu data *warehouse*, kita dapat menggunakan cara antara lain:

a. Cara yang paling sederhana adalah menyajikan data *warehouse* pada rentang waktu tertentu, misalnya antara 5 sampai 10 tahun ke depan.
b. Cara yang kedua, dengan menggunakan variasi/perbedaan waktu yang disajikan dalam data *warehouse* baik implicit maupun explicit secara explicit dengan unsur waktu dalam hari, minggu, bulan dsb. Secara implicit misalnya pada saat data tersebut diduplikasi pada setiap akhir bulan, atau per tiga bulan. Unsur waktu akan tetap ada secara implisit didalam data tersebut.
c. Cara yang ketiga,variasi waktu yang disajikan data *warehouse* melalui serangkaian snapshot yang panjang. Snapshot merupakan tampilan dari sebagian data tertentu sesuai keinginan pemakai dari keseluruhan data yang ada bersifat read-only.

4. Non-Volatile

Karakteristik keempat dari data *warehouse* adalah non-volatile,maksudnya data pada data *warehouse* tidak di-*update* secara *real time* tetapi di *refresh* dari sistem operasional secara reguler. Data yang baru selalu ditambahkan sebagai suplemen bagi database itu sendiri dari pada sebagai

sebuah perubahan. Database tersebut secara kontinyu menyerap data baru ini, kemudian secara incremental disatukan dengan data sebelumnya.

Berbeda dengan database operasional yang dapat melakukan update,insert dan delete terhadap data yang mengubah isi dari database sedangkan pada data *warehouse* hanya ada dua kegiatan memanipulasi data yaitu loading data (mengambil data) dan akses data (mengakses data *warehouse* seperti melakukan query atau menampilan laporan yang dibutuhkan, tidak ada kegiatan updating data).

2.2.2.2 Kegunaan Data *Warehouse*

Berdasarkan pengertian data *warehouse* diatas, data *warehouse* diperlukan bagi para pengambil keputusan manajemen dari suatu organisasi/perusahaan. Dengan adanya data *warehouse*, akan mempermudah pembuatan aplikasi-aplikasi DSS(***Decision Support System***) dan EIS(***Executive Information System***) karena kegunaan dari data *warehouse* adalah khusus untuk membuat suatu database yang dapat digunakan untuk mendukung proses analisa(OLAP), mengambil keputusan, pembuatan laporan, penggalian informasi baru(Data Mining) dari banyak data dan proses executive informasi.

Beberapa kegunaan data *warehouse:*

a. Pembuatan laporan

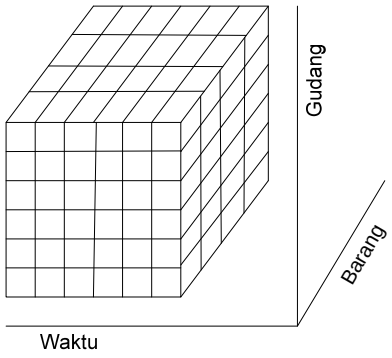
Pembuatan laporan merupakan salah satu kegunaan data *warehouse* yang paling umum dilakukan. Dengan menggunakan query sederhana didapatkan laporan perhari,perbulan, pertahun atau jangka waktu kapanpun yang diinginkan.

b. 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 multidimensi dan memungkinkan para pengguna menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL. Hal ini dimungkinkan karena pada konsep multi dimensi, maka data yang berupa fakta yang sama bisa dilihat dengan menggunakan fungsi yang berbeda. Fasilitas lain yang ada pada sofware OLAP adalah fasilitas rool-up dan drill-down. Drill-down adalah kemampuan untuk melihat detail dari suatu informasi dan roll-up adalah kebalikannya.

Operasi kunci dari OLAP adalah formasi kubus data atau representasi multidimensi dari data. Contoh representasi data dengan formasi kubus atau *ilustrasi cube*



Gambar 2.2 Ilustrasi Cube

Dari ilustrasi cube diatas dapat dilakukan analisa data dengan menggunakan beberapa operasi OLAP seperti *slice* dan *dice. Slice* adalah seleksi sekumpulan sel dari seluruh array multidimensi dengan menetapkan nilai spesifik untuk satu atau lebih dimensi. Sedangkan *dice* adalah seleksi dari subset sel dengan mentapkan interval nilai atribut.

c. Data mining

Data mining merupakan proses untuk menggali(mining) pengetahuan dan informasi baru dari data yang berjumlah banyak pada data *warehouse*, dengan menggunakan kecerdasan buatan (Artificial Intelegence), statistik dan matematika. Data mining merupakan teknologi yang diharapkan dapat menjembatani komunikasi antara data dan pemakainya.

Beberapa solusi yang diberikan data mining antara lain :

1. Menebak target pasar

Data mining dapat mengelompokkan (clustering) model-model pembeli dan melakukan klasifikasi terhadap setiap pembeli dan melakukan klasifikasi terhadap setiap pemebeli sesuai dengan karakteristik yang diinginkan.

2. Melihat pola beli dari waktu ke waktu

   Data mining dapat digunakan untuk melihat pola beli dari waktu ke waktu.

3. *cross-market analysis*

   Data mining dapat dimanfaatkan untuk melihat hubungan antara satu produk dengan produk lainnya.

4. Profil pelanggan

   Data mining bisa membantu pengguna untuk melihat profil pembeli sehingga dapat diketahui kelompok pembeli tertentu cenderung kepada suatu produk apa saja.

5. Informasi *summary*

   Data mining dapat membuat laporan summary yang bersifat multi dimensi dan dilengkapi dengan informasi statistik lainnya.

d. 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.

2.2.2.3 Konsep Dasar Data *Warehouse*

Data*warehouse* adalah kumpulan macam-macam data yang *subject oriented, integrated, time variant, dan nonvolatile* dalam mendukung proses pembuatan keputusan. Inmon and Hackathorn (1994).

Data*warehouse* sering diintegrasikan dengan berbagai sistem aplikasi untuk mendukung proses laporan dan analisis data dengan menyediakan data histori, yang menyediakan infrastruktur bagi EIS dan DSS.

a. *Subject Oriented*

Data*warehouse* diorganisasikan pada subjek-subjek utama, seperti pelanggan, barang/ produk, dan penjualan. Berfokus pada model dan analisis pada data untuk membuat keputusan, jadi bukan pada setiap proses transaksi atau bukan pada OLTP. Menghindari data yang tidak berguna dalam mengambil suatu keputusan.

b. *Integrated*

Dibangun dengan menggabungkan/menyatukan data yang berbeda. *relational databse, flat file, dan on-line transaction record.* Menjamin

konsistensi dalam penamaan, struktur pengkodean, dan struktur atribut diantara data satu sama lain.

c. Data*warehouse time variant*

Data disimpan untuk menyediakan informasi dari perspektif *historical*, data yang tahun-tahun lalu/ 4-5 thn. Waktu adalah elemen kunci dari suatu data*warehouse*/ pada saat pengcapture-an.

d. Non Volatile

Setiap kali proses perubahan, data akan di tampung dalam tiap-tiap waktu. Jadi tidak di perbaharui terus menerus. Data*warehouse* tidak memerlukan pemrosesan transaksi dan recovery. Hanya ada dua operasi *initial loading of data* dan *access of data*.

Data*warehouse* bukan hanya tempat penyimpanan data, Data*warehouse* adalah *Business Intelligence tools*, *tools to extract*, merubah (*transform*) dan menerima data (*load*) ke penyimpanan (*repository*) serta mengelola dan menerima metadata.

2.2.2.4 Proses ETL(Extraction, Transformation, Loading)

Tiga fungsi utama yang perlu dilakukan untuk membuat data siap digunakan pada data*warehouse* adalah extraction, transformation dan loading. Ketiga fungsi ini terdapat pada staging area.

Pada data staging ini, disediakan tempat dan area dengan beberapa fungsi seperti data cleansing, change, convert, dan menyiapkan data untuk disimpan serta akan digunakan dalam data *warehouse*

a. Extraction

Data Extraction adalah proses pengambilan data yang diperlukan dari sumber data *warehouse* dan selanjutnya dimasukkan pada staging area untuk diproses pada tahap berikutnya . Pada fungsi ini banyak berhubungan dengan berbagai tipe sumber data seperti: Format data, mesin yang berbeda, software dan arsitektur yang tidak sama. Sehingga sebelum proses ini dilakukan, sebaiknya perlu didefinisikan requirement terhadap sumber data yang yang akan digunakan untuk proses berikutnya.

b. Transformation

Pada kenyataannya, pada proses transaksional data disimpan dalam berbagai format sehingga jarang kita temui data yang konsisten antara aplikasi-aplikasi yang ada. Transformasi data ditujukan untuk mengatasi masalah ini. Dengan proses transformasi data ini, kita melakukan standarisasi terhadap data pada satu format yang konsisten. Beberapa contoh ketidakkonsistenan data tersebut dapat diakibatkan oleh tipe data yang berbeda, data length dan lain sebagainya.

c. Load

Data load adalah memindahkan data ke data*warehouse*. Ada dua loading data yang dilakukan pada data*warehouse*. Pertama adalah inisial load, proses ini dilakukan pada saat telah selesai mendisain dan membangun data *warehouse.* Data yang dimasukkan akan sangat besar dan memakan waktu yang relati lebih lama. Kedua Incremental load, dilakukan ketika data *warehouse* telah dioperasikan. Incremental load ini dapat dilakukan sesuai dengan sistem yang dibangun.

2.2.2.5 Arsitektur Data *warehouse*

Maksud dari Arsitektur Data*warehouse* adalah bagaimana data*warehouse* dibangun, arsitektur tidak ada yang benar dan salah tetapi suatu arsitektur dibangun tergantung situasi dan kondisi. Arsitektur data*warehouse* akan berpengaruh pada penggunaan dan pemeliharaan. Lapisan-lapisan arsitektur data*warehouse* :

1. Operational database layer / Lapisan basis data operasional
    a. Sumber data (source) untuk data*warehouse*
    b. Data lengkap, Data hari ke hari
    c. Mempunyai nilai saat ini/ data berarti.
    d. Tingkat kemungkinan data besar.
2. Data Access Layer/ Lapisan Akses Data
    a. Tools untuk mengekstrak, mengubah dan mengambil (load) data.

b. Meliputi karakteristik data*warehouse*.

3. Metadata Layer / Lapisan Metadata

a. File data tersimpan / Direktori

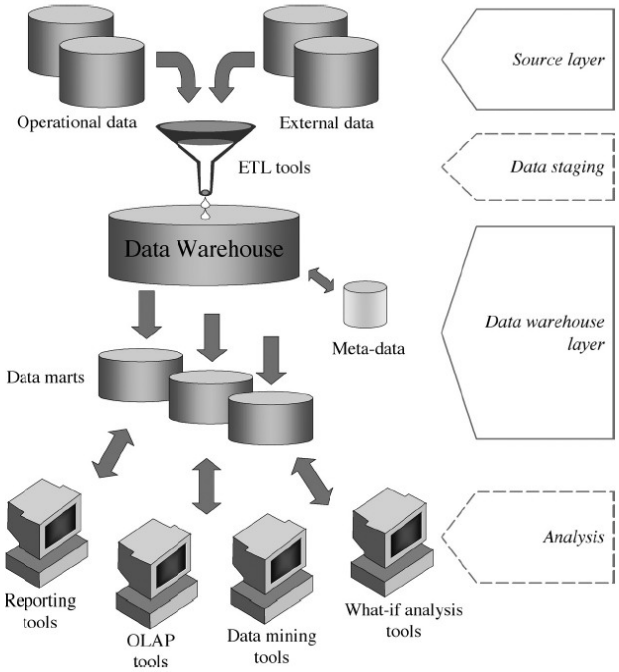
b. Lebih detil dari direktori data sistem, maksudnya lebih mendalam dari file data yang tersedia sebelumnya.

c. Ada pentunjuk untuk keseluruhan *warehouse* dan ada petunjuk data yang dapat diakses report khusus untuk di analisis.

4. Informational access layer (lapisan akses informasi)

a. Akses data dan juga tool untuk laporan dan analisis.

b. Tools Business Intelligence masuk ke tahap ini.



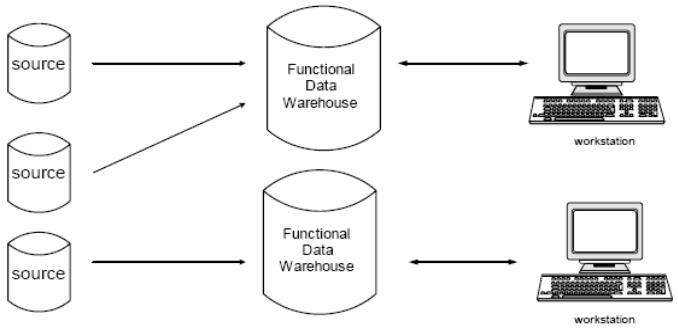
Gamabar 2.3 Arsitektur Data*warehouse*

2.2.2.6 Jenis Dasar Data *Warehouse*

Jenis data *warehouse* yang digunakan oleh perusahaan akan berbeda-beda, data *warehouse* yang dibangun harus disesuaikan dengan sumber data dan kebutuhan perusahaan tersebut. Berikut jenis dasar data *warehouse*:

1. Functional Data *Warehouse* (*Data Warehouse Fungsional)*

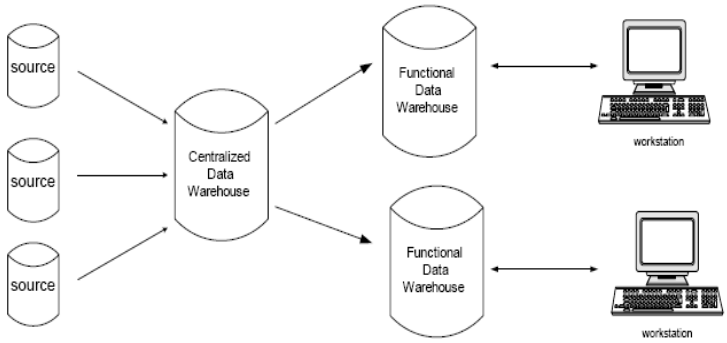
Data base yang digunakan jenis data*warehouse* ini adalah database yang diperoleh dari kegiatan sehari-hari. Data *warehouse* dibuat lebih dari satu dan dikelompokkan berdasar fungsi-fungsi yang ada di dalam perusahaan seperti fungsi keuangan(*financial),*marketing,personalia dan lain-lain. Keuntungan dari bentuk data *warehouse* seperti ini adalah, sistem mudah dibangun dengan biaya relatif murah sedangkan kerugiannya adalah resiko kehilangan konsistensi data dan terbatasnya kemampuan dalam pengumpulan data bagi pengguna.



Gambar 2.4 Data *Warehouse* Fungsional

2. Centralized Data*warehouse* (*Data Warehouse Terpusat)*

Bentuk ini terlihat seperti bentuk data *warehouse* fungsional, namun terlebih dahulu sumber data dikumpulkan dalam satu tempat terpusat, kemudian data disebar ke dalam fungsinya masing-masing, sesuai kebutuhan persuhaan. Data *warehouse* terpusat ini, biasa digunakan oleh perusahaan yang belum memiliki jaringan eksternal. Keuntungan dari bentuk ini adalah data benar-benar terpadu karena konsistensinya yang tinggi sedang kerugiannya adalah biaya yang mahal serta memerlukan waktu yang cukup lama untuk membangunnya.



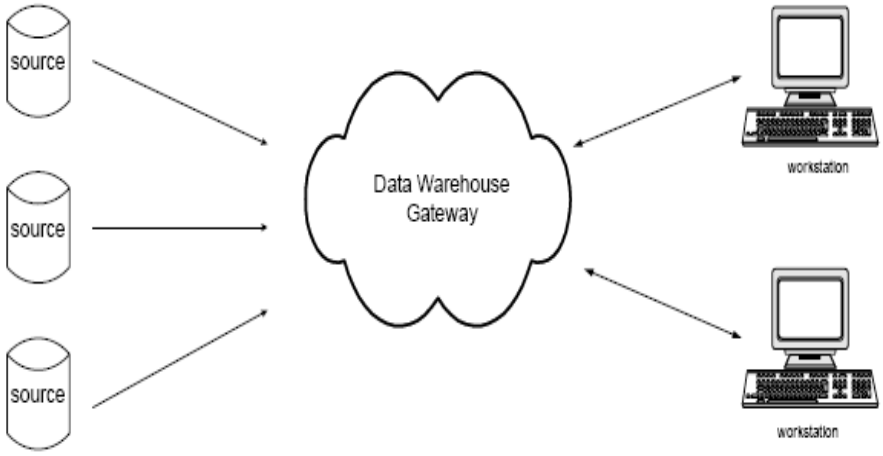
Gambar 2.5 ***Data Warehouse Terpusat***

3. Distributed Data *Warehouse* (*Data Warehouse terdistribusi)*

Pada data *warehouse* terdistribusi ini, digunakan gateway yang berfungsi sebagai jembatan penghubung antara data *warehouse* dengan workstation yang menggunakan sistem beraneka ragam. Dengan sistem

terdistribusi seperti ini memungkinkan perusahaan dapat mengakses sumber data yang berada diluar lokasi perusahaan(eksternal).

Keuntungannya adalah data tetap konsisten karena sebelum data digunakan data terlebih dahulu di sesuaikan atau mengalami proses sinkronisasi. Sedangkan kerugiannya adalah lebih kompleks untuk diterapkan karena sistem operasi dikelola secara terpisah juga biaya nya yang paling mahal dibandingkan dengan dua bentuk data *warehouse* lainnya.



Gambar 2.6 *Data Warehouse terdistribusi*

2.2.2.7 Metodologi Perancangan Database untuk *Data Warehouse*

Menurut Kimball ada sembilan tahap metodologi dalam perancangan database untuk *data warehouse*, yaitu :

Langkah 1 : Pemilihan proses

a. Data mart yang pertama kali dibangun haruslah data mart yang dapat dikirim tepat waktu dan dapat menjawab semua pertanyaan bisnis yang penting
b. Pilihan terbaik untuk data mart yang pertama adalah yang berhubungan dengan *sales*, misal *property sales*, *property leasing*,*property advertising.*

Langkah 2 : Pemilihan sumber

a. Proses pemilihab secara pasti apa yang diwakili atau direpresentasikan oleh sebuah tabel fakta.
b. Misal, jika sumber dari sebuah tabel fakta properti sale adalah properti sale individual maka sumber dari sebuah dimensi pelanggan berisi rincian pelanggan yang membeli properti utama

Langkah 3 : Mengidentifikasi dimensi

a. Set dimensi yang dibangun dengan baik, memberikan kemudahan untuk memahami dan menggunakan data mart
b. Dimensi ini penting untuk menggambarkan fakta-fakta yang terdapat pada tabel fakta
c. Misal, setiap data pelanggan pada tabel dimensi pembeli dilengkapi dengan id_pelanggan,no_pelanggan,tipe_pelanggan,tempat_tinggal, dan lain sebagainya.

d. Jika ada dimensi yang muncul pada dua data mart,kedua data mart tersebut harus berdimensi sama,atau paling tidak salah satunya berupa subset matematis dari yang lainnya.
e. Jika sebuah dimensi digunakan pada dua data mart atau lebih,dan dimensi ini tidak disinkronisasi,maka keseluruhan *data warehouse* akan gagal, karena dua data mart tidak bisa digunakan secara bersama-sama

Langkah 4 : Pemilihan fakta

a. Sumber dari sebuah tabel fakta menentukan fakta mana yang bisa digunakan dalam data mart.
b. Semua fakta harus diekspresikan pada tingkat yang telah ditentukan oleh sumber

Langkah 5 : Menyimpan pre-kalkulasi di tabel fakta

Hal ini terjadi apabila fakta kehilangan statement

Langkah 6 : Melengkapi tabel dimensi

a. Pada tahap ini kita menambahkan keterangan selengkap-lengkapnya pada tabel dimensi
b. Keterangannya harus bersifat intuitif dan mudah dipahami oleh pengguna

Langkah 7 : Pemilihan durasi database

Misalnya pada suatu perusahaan asuransi, mengharuskan data disimpan selama 10 tahun atau lebih

Langkah 8 : Menelusuri perubahan dimensi yang perlahan

Ada tiga tipe perubahan dimensi yang perlahan, yaitu :

1. Atribut dimensi yang telah berubah tertulis ulang
2. Atribut dimensi yang telah berubah menimbulkan sebuah dimensi baru
3. Atribut dimensi yang telah berubah menimbulkan alternatif sehingga nilai atribut lama dan yang baru dapat diakses secara bersama pada dimensi yang sama.

Langkah 9 : Menentukan prioritas dan mode *query*

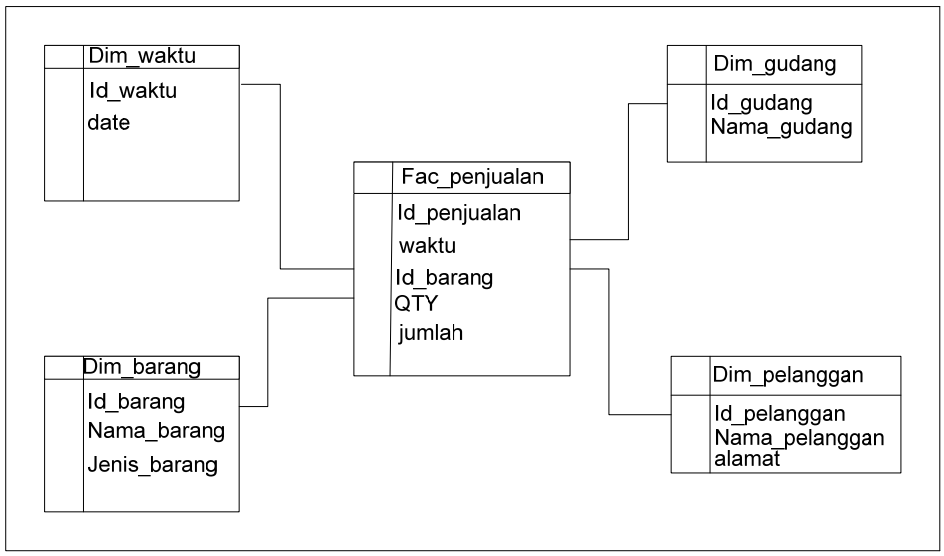
Pada tahap ini lebih menggunakan perancangan fisik.

2.2.2.8 Skema Data *Warehouse*

Skema yang sering digunakan pada *data warehouse* adalah skema bintang atau *snowflake*, kedua skema ini sangat mudah dimengerti dan sesuai dengan kebutuhan bisnis, mendukung *query* sederhana dan menyediakan performa *query* yang superior dengan meminimalisasi tabel-tabel join.

2.2.2.8.1 Skema Bintang (Star Schema)

Skema bintang merupakan struktuk logikal yang memiliki tabel fakta yang terdiri atas data faktual ditengahnya, dan dikelilingi oleh tabel-tabel dimensi yang berisi referensi data.

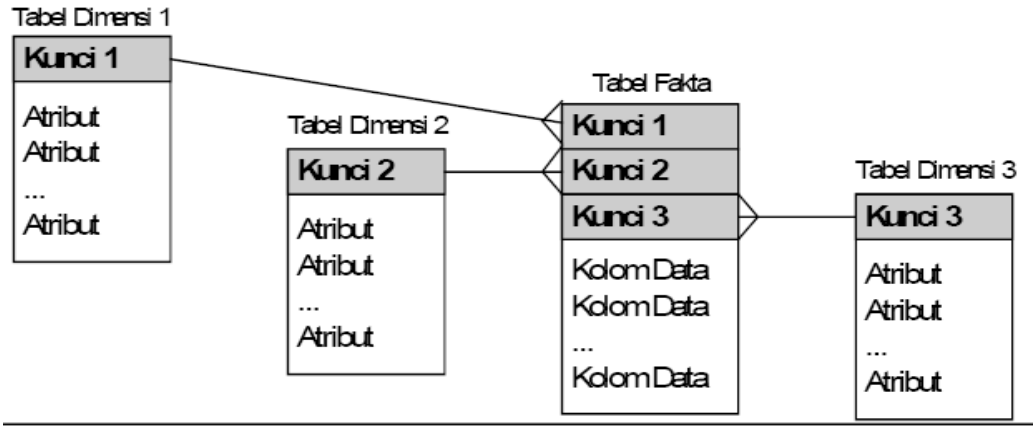


Gambar 2.7 Skema Bintang

Jenis-jenis Skema Bintang

1. Skema bintang sederhana

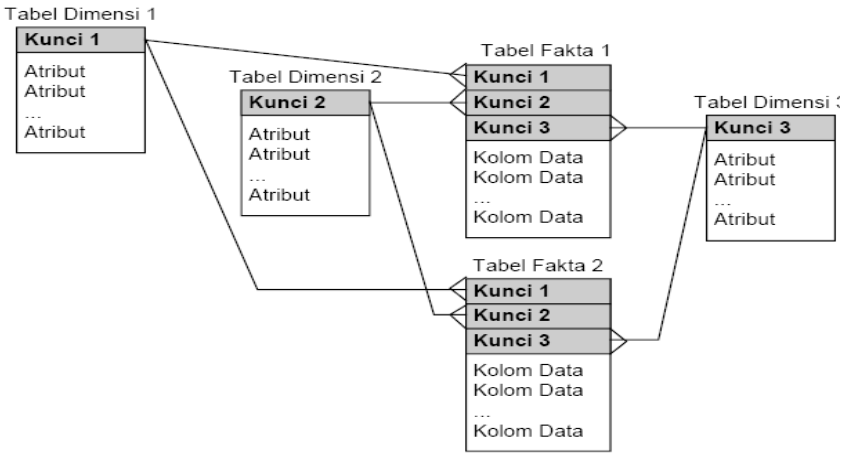
   Dalam skema ini, setiap table harus memiliki *primary key* yang terdiri dari satu kolom atau lebih. *Primary key* dari table fakta terdiri dari satu atau lebih *foreign key.Foreign key* merupakan *primary key* pada table lain.



Gambar 2.8 Skema Bintang sederhana

2. Skema bintang dengan banyak table fakta

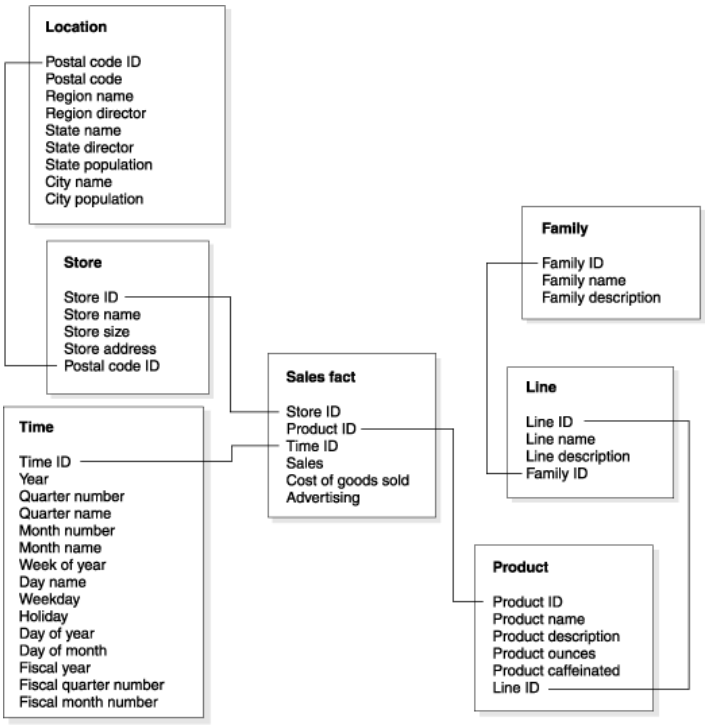
Skema bintang juga bisa terdiri dari satu atau lebih table fakta. Dikarenakan karena table fakta tersebut ada banyak, misalnya disamping penjualan terdapat table fakta *forecasting* dan *result.* Walaupun terdapat lebih dari satu table fakta, mereka tetap menggunakan table dimensi bersama-sama.



Gambar 2.9 Skema Bintang sederhana

2.2.2.8.2 Snowflake Schema

Merupakan varian dari skema bintang dimana table-table dimensi tidak terdapat data yang di denormalisasi. Dengan kata lain satu atau lebih table dimensi tidak bergabung secara langsung kepada table fakta tapi pada table dimensi lainnya.



Gambar 2.10 Snowflake schema

2.2.2.9 Tujuan Data*warehouse*

Data*warehouse* biasanya dimiliki oleh perusahaan yang sudah besar, yang memiliki banyak cabang, data yang banyak dan tentunya struktur organisasi yang kompleks.

Perusahaan yang memiliki banyak transaksi, yang memiliki banyak cabang, yang mengakibatkan data-data barang perusahaan tersebar dilokasi yang berbeda, system operasi yang berbeda, bahkan di basisdata yang berbeda. Sehingga dalam pengambilan sebuah keputusan pihak perusahaan membutuhkan akses ke semua sumber data pada tiap-tiap cabang dan ini tidak efisien. Dari permasalahan ini,

Data*warehouse* hadir sebagai solusi yang kompetitif, efisien untuk pengambilan keputusan di perusahaan.

Tujuan Akhir menggunakan Data*warehouse*

a. Menyediakan data organisasi yang mudah diakses oleh manager.

b. Data yang berada di data*warehouse* bersifat konsisten, dan merupakan kebenaran.

c. Data*warehouse* merupakan tempat, dimana data yang telah digunakan di publikasikan.

d. Kualitas data di data*warehouse* dapat diandalkan.

2.2.3 konsep Rekayasa Perangkat Lunak

2.2.3.1 Pengertian Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak adalah aplikasi dari serbuah pendekatan kuantifiabel, disiplin dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak. Usah yang berhubungan dengan rekayasa perangkat lunak dapat dikategorikan ke dalam tiga fase umum dengan tanpa memperdulikan area aplikasi, ukuran proyek, atau kompleksitasnlya, yaitu :

1. Fase Defenisi (Definition Phase)

   Fase ini berfokus pasa “apa” (what); dimana, pada definsi ini pengembang perangkat lunak harus mengidentifikasi informasi apa yang akan di proses,

fungsi dan unjuk kerja apa yang dibutuhkan, tingkah laku system seperti apa yang di harapkan, antarmuka apa yang akan dibangun, batasan perancanagan serta criteria validasi untuk mendifinisikan system yang sukses. Tugas teknis yang harus selalu ada dalam fase ini yaitu rekayasa system atau informasi, perencanaan proyek perangkat lunak, serta analisis kebutuhan.

2. Fase Pengembangan (Development phase)

   Fase ini berfokus pada “bagaimana” (how), yaitu dimana selama masa pengembangan perangkat lunak, teknisi harus mendefinisikan bagaimana data dikonstruksikan, bagaimana fungsi-fungsi diimplementasikan sebagai sebuah arsetktur perangkat lunak, bagaimana detail prosedur akan diimplementasikan, bagaimana antarmuka dikarakterisasi, bagaimana rancangan diterjemahkan ke dalam bahasa pemograman serta bagaimana penguian akan dilakukan. Tugas teknis harus selalu ada dalam fase ini yaitu rancangan perangkat lunak, pemunculan kode, pengujian perangkat lunak.

3. Fase Pemeliharaan (maintenance Phase)

   Fase ini berfokus pada “perubahan” (change), yang dihubungkan dengan koreksi kesalahan, penyesuaian yang dibutuhkan ketika lingkungan perangkat lunak berkembang, serta perubahan kebutuhan pelanggan. Fase ini mengaplikasikan kembali langkah-langkah pada fase defenisi dan pengembangan namun semuanya teteap bergantung pada konteks perangkat lunak yang ada.

Untuk menyelesaikan masalah aktual di dalam sebuah setting industry, rekayasa perangkat lunak atau tim perekayasa harus menggabungkan strategi pengembangan yang melingkupi lapisan proses, metode, dan alat-alat bantu serta fase-fase generic. Strategi ini sering diacukan sebagai model proses atau paradigma rekayasa perangkat lunak. Model proses atau rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat-alat bantu yang akan dipakai, dan control penyampaian yang dibutuhkan

Model proses untuk rekayasa perangkat lunak dibagi menjadi empat model yaitu :

1. Model Sekuensial Linier
2. Model Prototipe
3. Model RAD
4. Model Proses Perangkat Lunak Evlusioner

2.2.3.2 Model Sekuensial (*Waterfall*)

Metode yang di gunakan dalam pembuatan perangkat lunak ini mengacu pada paradigma model waterfall dengan langkah-langkah sebagai berikut:

a. *Engineering sistem* (rekayasa sistem) adalah menentukan kebutuhan-kebutuhan dari semua elemen sistem dan mengalokasikan suatu subset kedalam pembentukan sistem yang akan dibuat.

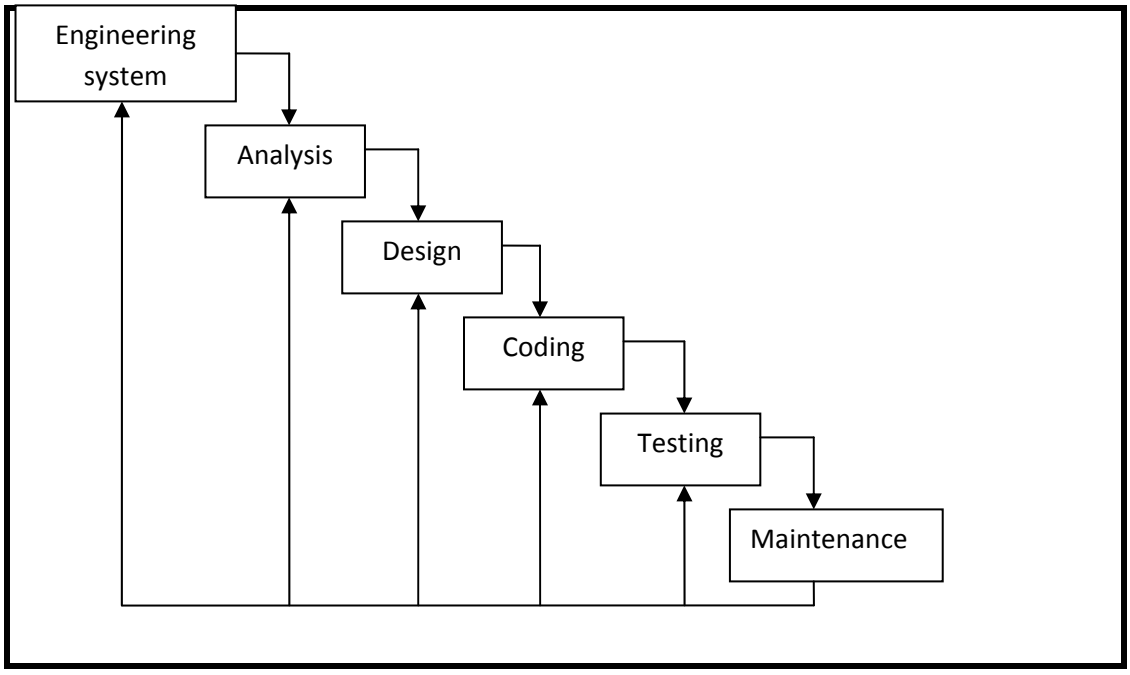
b. *Analysis* (analisis) adalah tahap pengumpulan kebutuhan untuk membentuk domain informasi dari sistem yang sedang berjalan dengan melakukan analisis.

c. *Design* (desain) adalah proses mengubah kebutuhan -kebutuhan menjadi bentuk karakteristik yang dimulai dengan penulisan program, design tadi harus diubah menjadi bentuk yang dimengerti sistem dengan berdasarkan analisis yang telah dilakukan.

d. *Coding* (pengkodean) adalah tahap penerjemah desain kedalam bahasa pemograman yang sesuai dengan kebutuhan.

e. *Testing* (pengujian) adalah tahap pengujian perangkat lunak yang dikembangkan dan menjamin bahwa masukan sesuai dengan hasil yang dibutuhkan.

f. *Maintenance* (pemeliharaan) adalah tahap perawatan sistem yang telah dikembangkan seperti perawatan perangkat lunak,perawatan perangkat keras dan media lain yang berhubungan dengan komputer. Pada tahap ini pula harus dijaga perfomance perangkat lunak agar berjalan dengan baik



Gambar 2.11 Paradigma Model *Waterfall*

2.2.4 Konsep Perancangan Sistem

Perancangan system secara umum adalah suatu tahap dimana di dalamnya terdapat identifikasi komponen-komponen system infomasi yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau user mengenai system yang baru. Sedangkan desain system secara terinci dimaksudkan untuk pembuat program computer dan ahli teknik lainnnya yang akan *mengimplementasikan* sistem.

Penggambaran dan rancangan model system infomasi secara logika dapat dibuat dalam bentuk diagaram konteks dan Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD).

2.2.4.1 Diagram Konteks

Diagram kontek merupakan alat-alat untuk struktur analisis. Pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan. Diagram kontek adalah kasus khusus dari DFD atau bagan dari DFD yang berfungsi memetakan modul lingkungan yang dipersentasika dengan lingkaran tunggal yang mewakili keseluruhan sistem.

2.2.4.2 Diagaram Alir Data

Diagram alir data atau *Data Flow Diagram* (DFD) adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagaram konteks yang menjelaskan cecara umum suatu system atau batasan system aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi menjadi level 1 dan selanjutnya sampai system tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut :

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas
2. Penurunan dilakukan apabila memang diperlukan
3. Tidak semua bagian dari system harus ditunjukan dengan jumlah level yang sama.

Symbol-simbol yang digunakan pada diagram alir data ata data flow diagram antara lain :

a. Entitas Luar (*external entity*)
b. Arus data (*Data flow*)
c. Proses (*proccess*)
d. Simpanan data (*data store*)

2.2.5 Pengertian Basis data

Basis data terdiri dari kata basis dan data. Basis data dapat diartikan gudang atau tempat bersarang dan data yang berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebaginya yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi atau kombinasinya. Jadi dapat disimpulkan bahwa basis data merupakan kumpulan data yang saling

berhubungan yang disimpan secara berama sedemikian rupa dantanpa pengulangn (redudansi) yang tidak perlu, untuk memnuhi berbagai kebutuhan. Atau bisa diartikan sebagai kumpulan file,table, arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.2.5.1 Konsep Dasar Basis Data

Basis data (*database*) dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis data menunjukkan suatu kumpulan data yang dipakai dalam system informasi disebut sistem basis data(*database system*)

2.2.6.2 Basis Data Relasional

Konsep sebuah database adalah terdiri atas tabel-tabel yang terorganisasi. Tabel-tabel tersebut dapat saling berelasi untuk menghasilkan suatu infomasi, untuk mengakses data yang ada dalam tabel-tabel tersebut digunakan sebuah perintah SQL(Structured Query language)

2.2.5.3 Perancangan Basis Data

Perancangan basis data dibutuhkan agar diperoleh suatu sistem yang sesuai dengan apa yang diinginkan, dimana dapat melalui tahapan berikut :

2.2.5.4 ERD *(Entity Relationship Diagram)*

Entity Relationship Diagram adalah suatu model atau teknik pendekatan yang dapat menyatakan atau menggambarkan hubungan *entity* didalam sebuah sistem, dimana hubungan tersebut dinyatakan sebagai *one-to-one*, *one-to-many* dan *many-to-many*. *Entity relationship* bertujuan untuk menentukan objek-objek data (*Entity*) dan *relationship* (hubungan) yang ada pada objek-objek tersebut.

Komponen *Entity Relationship Diagram* :

1. *Entity*

   Entity adalah suatu yang dapat dibedakan atau diidentifikasikan secara unik, objek disini sifatnya berupa orang, tempat, peristiwa atau konsep yang informasinnya direkam.

2. Atribut

   Atribut adalah kumpulan dari beberapa elemen data yang membentuk suatu entity. Atribut merupakan ciri sebutan atau karakteristik yang dapat mewakili suatu simbol yang digunakan untuk menggambarkan atribut pada *entity relationship diagram.*

3. *Relationship*

   Relationship adalah relasi atau hubungan yang terjadi diantara beberapa entity. Simbol yang digambarakan untuk menggambarkan *relationship* pada teknik *entity relationship diagram* adalah berupa dimendi (Belah ketupat).

Setiap *relationship* mempunyai derajat *relationship*, yaitu tingkat atau hubungan yang terdiri dari tiga derajat, yaitu :

a. *Unary* (derajat 1)

   Suatu *relationship* yang dihubungkan dengan suatu *entity* set, dimana penghubungnya ada dua.

b. *Binary* (derajat 2)

   Relationship derajat dua adalah *relationship* yang biasa terjadi yaitu dua *entity* set dihubungkan dengan satu *entity relationship*.

c. *Ternary* (derajat 3)

   Relationship derajat tiga.

Relationship diagram juga mempunyai *Cardinalitas* yaitu batasan pada jumlah *entity* yang terdiri dari:

1. Relasi satu ke satu (*one-to-one*)

   Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa satu berbanding satu.

2. Relasi satu ke banyak (*one-to-many*)

   Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak atau banyak berbanding satu.

3. Relasi Banyak ke Banyak (*many-to-many*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak.

2.2.5.5 Kamus Data

Kamus data adalah suatu daftar atau kamus dari seluruh elemen-elemen data yang diperlukan oleh suatu sistem. Kamus data dibuat berdasarkan arus data yang terdapat pada DFD. Arus data dalam DFD bersifat global sehingga hanya dapat ditunjukkan nama arus datanya saja dan keterangan lebih lanjut mengenai struktur arus data dapat dilihat dari kamus data.

2.2.6.6 *Database Management System* (DBMS)

Database Management System (DBMS) diartikan sebagai suatu program komputer yang digunakan untuk memasukan, mengubah, menghapus, memanipulasi dan memperoleh data atau informasi dengan praktis dan efisien. [7]

Dibandingkan dengan sistem yang berbasis kertas, DBMS memiliki empat keunggulan :

2. Kepraktisan. Sistem yang berbasis kertas akan menggunakan kertas yang sangat banyak untuk menyimpan informasi, sedangkan DBMS menggunakan media penyimpan sekunder yang berukuran kecil tetapi padat informasi.

3. Kecepatan. Mesin dapat mengambil atau mengubah data jauh lebih cepat dari pada manusia.
4. Mengurangi kejemuan. Orang cenderung menjadi bosan kalau melakukan tindakan-tindakan berulang yang menggunakan tangan.
5. Kekinian. Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat.

Di dalam lingkungan basis data, data lebih mudah digunakan pada beberapa DBMS tersedia fasilitas Query yang memudahkan pengguna untuk memperoleh informasi. Pengguna yang tidak memiliki kemampuan pemograman pun dengan mudah bisa mengguankan fasilitas query tersebut.

2.2.6 Perangkat Lunak Pendukung

Perangkat lunak *(software)* pendukung sangatlah dibutuhkan dalam membangun sistem aplikasi ini, karena sistem aplikasi yang akan dibangun ini membutuhkan beberapa program aplikasi yang digunakan untuk menghasilkan program aplikasi yang lengkap yang sesuai dengan yang dibutuhkan oleh pengguna. Adapun program aplikasi yang digunakan dalam pembangunan perangkat lunak ini adalah *Microsoft Delphi 6 MySQL (XAMPP).*

2.2.6.1 Tinjauan software Borland Delphi 6

Borland Delphi merupakan bahasa pemrograman yang menggunakan fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada

kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrograman terstruktur. Keunggulan lain dari Delphi seperti program aplikasi lain berbasis windows.

Dalam hal pemrograman database, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format database yang dimiliki Delphi adalah format database *paradox, Ms.Acces, ODBC, SYBASE, Oracle* dan lain-lain.

a. IDE Delphi

IDE (Integrated Development Environment) adalah bagian dari Delphi yang digunakan untuk menciptakan aplikasi, dengan IDE inilah para programmer secara visual merancang tampilan untuk para user (antarmuka pengguna) dan menulis listing program (kode). IDE dalam pemrograman Delphi 5 terdiri dari beberapa bagian yang masing-masing memiliki komponen-komponen dan fungsi-fungsi tertentu. Bagian-bagian tersebut diantaranya adalah Menubar, Toolbar/Speedbar, Component Palette, Jendela Form, Jendela Unit, Object Treeview dan Object Inspector. Untuk keterangan lebih lanjut dibawah ini terdapat penejelasan tentang komponen-komponen serta fungsi-fungsi tersebut.

b. Aplikasi dan Form

Aplikasi merupakan penerapan, pengimplementasi suatu permasalahan pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan permasalahan tersebut, sehingga menjadi suatu bentuk program tanpa menghilangkan nilai-nilai dasar dari permasalahan yang ada. Aplikasi dapat dibedakan menjadi aplikasi windows dan aplikasi konsol. Aplikasi windows adalah aplikasi yang berjalan pada windows. Aplikasi Non windows contohnya yang berjalan pada Dos yang biasa disebut aplikasi konsol. Secara umum, sebuah aplikasi paling sedikit melibatkan sebuah form. Namun biasanya sebuah aplikasi melibatkan banyak form. Sebuh form umumnya melibatkan komponen lain. Kotak Combobox dan Tombol Radiobutton merupakan contoh komponen. Tetapi tidak semua komponen terlihat secara visual, komponen yang terlihat secara visual biasa disebut dengan control. Pada Delphi, sebuah aplikasi akan diletakkan pada sebuah proyek. Sebuah proyek dapat membawahi sejumlah form. Form merupakan inti dari setiap aplikasi pada Delphi karena dianggap sebagai dasar aplikasi windows.

2.2.6.2 Pengertian MySQL

MySQL adalah sebuah perangkat lunak yang digunakan untuk mengolah database pada konsep *client-server*, dimana MySQL ini didistribusikan secara gratis. Dengan menggunakan database ini, data semakin aman dan berdaya guna. Database ini juga banyak dipakai pada web database sehingga data semakin terintegrasi antara database dekstop dengan database web. Untuk menggunakan database MySQL harus menginstalasinya dahulu ke komputer.

Keistimewaan MySQL

Sebagai server database dengan konsep database modern, MySQL memiliki keistimewaan. Beberapa keistimewaan dimiliki MySQL sebagai berikut :

1). Portability

Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-Unix, dan lain-lain.

2). Open Source

MySQL merupakan database open source (gratis), di bawah lisensi GPL sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.

3). Multiuser

MySQL merupakan database yang dapat digunakan untuk menangani beberapa user dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan pula.

4). Performace Tuning

MySQL mempunyai kecepatan yang cukup baik dalam menangani query-query sederhana, serta mampu memproses lebih banyak SQL per satuan waktu.

5). Column Type

Database MySQL didukung dengan tipe data yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set serta enum.

6). Command dan Functions

MySQL server memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

7). Security

Sistem Security pada MySQL mempunyai beberapa lapisan sekuritas seperti tingkatan subnetmask, hostname, dan izin akses user dengan sistem perizinan yang mendetil serta password terenkripsi.

8). Scalability dan Limits

MySQL mempunyai kemampuan menangani database dalam skala cukup besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu dapat menampung indeks sampai 32 indeks pada tiap tabelnya.

9). Connectivity

Adanya kemampuan MySQL melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).

10).Localization

Adanya kemampuan dalam mendeteksi kesalahan (error code) pada client menggunakan lebih dari dua puluh bahasa.

11).Interface

MySQL memiliki interface terhadap berbagai aplikasi dan bahasa pemograman menggunakan fungsi API (Application Programming Interface).

12).Clients dan Tools

Database MySQL dilengkapi berbagai tools yang dapat digunakan untuk administrasi database.

13).Struktur Tabel

MySQL memiliki struktur tabel cukup baik serta cukup fleksibel, misalnya ketika menangani alter table.