

BAB 2

LANDASAN TEORI

2.1 Teori-teori Dasar

2.1.1 Pengertian Sistem

Untuk dapat membuat sebuah analisa dan perancangan suatu sistem, kita harus lah sebaik nya mengerti terlebih dahulu apa itu sistem, apa pengertian dari suatu sistem. Oleh karena itu dilakukan lah pembahasan mengenai sistem itu sendiri.

Menurut Mathiassen,et.al.(2000,p9), “Sistem adalah kumpulan komponen yang mengimplementasikan permintaan model, fungsi, dan *interface*.”

Menurut Hall (2001,p5), ”Sistem adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (*interrelated*) atau subsistem-subsistem yang bersatu untuk mencapai suatu tujuan yang sama.”

Menurut O’Brien (2003,p8),”Sistem adalah kumpulan komponen-komponen yang saling berhubungan yang saling bekerja sama untuk mendapatkan tujuan utama dengan penerimaan *input* dan

pengeluaran *output* dalam suatu organisasi yang melibatkan transformasi proses.”

Menurut McLeod (2001,p11),” Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.”

Jadi pengertian sistem adalah sekelompok elemen-elemen yang saling berkaitan satu dengan lainnya yang saling bekerja sama untuk mencapai tujuan tertentu.

2.1.2 Pengertian Informasi

Kita juga perlu mengetahui apa itu arti informasi itu sendiri. Menurut McLeod (2001,p5),”Informasi adalah data yang telah diproses atau data yang memiliki arti.”

Jadi pengertian informasi adalah data-data yang telah diolah sehingga mempunyai suatu manfaat bagi penerimanya dalam pengambilan keputusan,pengontrolan dan analisi masalah

Informasi yang dihasilkan harus memiliki criteria sebagai berikut :

- Informasi harus lengkap, arti nya informasi meliputi semua data yang berhubungan dengan kebutuhan pemakai.
- Informasi harus tepat waktu, arti nya informasi yang disajikan harus tepat waktu pada saat dibutuhkan.
- Informasi harus akurat, arti nya informasi yang dihasilkan harus bebas dari kesalahan.

- Informasi harus relevan, artinya informasi yang disajikan harus sesuai dengan kebutuhan pemakai

2.1.3 Pengertian Sistem Informasi

menurut Hall (2001,p7),”Sistem Informasi adalah sebuah rangkaian prosedur formal dimana data dikumpulkan, diproses menjadi informasi, dan didistribusikan kepada para pemakai.”

Menurut Turban et.al (2001,p17),”Sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan mendistribusikan informasi untuk tujuan tertentu.”

Menurut O’Brien (2003,p7), ”Sistem informasi adalah kombinasi diantara orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang dikumpulkan, diubah, dan disebarkannya informasi di dalam suatu organisasi.”

Jadi, pengertian sistem informasi adalah sekumpulan kerangka kerja yang berfungsi mengolah data menjadi informasi dalam mencapai tujuan perusahaan.

2.1.4 Tujuan Sistem Informasi

Tujuan sistem informasi yang spesifik dalam tiap-tiap perusahaan dapat berbeda. Namun demikian, terdapat tiga tujuan utama bagi semua sistem menurut Hall (2001, p18), yaitu:

- Untuk mendukung fungsi kepengurusan (*Stewardship*) manajemen. Kepengurusan merujuk ke tanggung jawab manajemen untuk mengatur sumber daya perusahaan secara benar. Sistem informasi menyediakan informasi tentang kegunaan sumber daya ke pemakai eksternal melalui laporan keuangan tradisional dan laporan-laporan yang diminta lainnya. Secara internal, pihak manajemen menerima informasi kepengurusan dari berbagai laporan pertanggung jawaban.
- Untuk mendukung pengambilan keputusan manajemen. Sistem informasi memberikan para manajer informasi yang mereka perlukan untuk melakukan tanggung jawab pengambilan keputusan.
- Untuk mendukung kegiatan operasi perusahaan hari demi hari. Sistem informasi menyediakan informasi bagi personel operasi untuk membantu mereka melakukan tugas mereka setiap hari secara efektif dan efisien.

2.1.5 Komponen Sistem Informasi

Menurut Turban et.al. (2001,p17), komponen dasar sistem informasi adalah sebagai berikut:

- Perangkat Keras (*Hardware*)

Sekumpulan peralatan seperti *processor, monitor, keyboard, printer* yang menerima data dan informasi, memproses. serta menampilkan nya

- Perangkat Lunak (*Software*)

Sekumpulan program komputer yang memungkinkan perangkat keras untuk memproses data.

- Basis Data (*Database*)

Kumpulan dari file, record, dan lain-lain yang terorganisasi dimana berguna untuk menyimpan data dan hubungannya.

- Jaringan (*Network*)

Suatu sistem yang terhubung dimana menyediakan penggunaan secara bersama-sama sumber daya antar komputer yang berbeda.

- Prosedur (*Procedures*)

Strategi, kebijakan, metode, dan aturan untuk menggunakan sistem informasi.

- Personil (*People*)

Merupakan elemen yang paling penting dalam sistem informasi yang terdiri dari mereka yang bekerja dengan sistem informasi itu sendiri atau menggunakan *output*.

2.1.6 Pengertian Data

Menurut O'Brien (2003,p13), data adalah fakta mentah atau observasi, umumnya tentang fenomena fisik atau transaksi bisnis. Sebagai contoh, sebuah peluncuran pesawat luar angkasa atau penjualan suatu mobil akan menciptakan banyak data yang menggambarkan kejadian tersebut. Secara lebih spesifik, data adalah ukuran objektif dari atribut (karakteristik) dari entitas (seperti orang, tempat, dan benda).

Menurut Turban et al (2001,p17), data adalah fakta mentah atau deskripsi dasar mengenai suatu benda, peristiwa, kegiatan dan transaksi yang ditangkap, direkam, disimpan dan diklasifikasikan, tetapi tidak diperuntukkan untuk menyimpan arti apapun.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa data adalah sebuah fakta atau pernyataan yang bersifat mentah, diterima apa adanya yang umumnya berisi tentang fenomena fisik atau transaksi fisik atau transaksi bisnis dan fakta tersebut berupa angka, kata-kata, atau gambar.

2.2 Teori Khusus

2.2.1 Pengertian Basis Data

Menurut Connolly et al (2002, p14), basis data adalah koleksi yang terbagi dari data yang berhubungan baik secara logical dan deskripsi dari data tersebut yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Database merupakan suatu tempat penyimpanan data yang besar yang dapat digunakan secara simultan oleh banyak user atau departmen.

Menurut Post, Gerald V (2005, p2), basis data adalah sekumpulan data yang disimpan dalam bentuk tertentu yang dirancang agar dapat digunakan oleh pengguna yang beragam

Dari dua definisi diatas dapat disimpulkan bahwa basis data adalah sekumpulan data yang berhubungan baik secara logical dan deskripsi yang dirancang untuk memenuhi kebutuhan yang bermacam-macam.

2.2.1.1 Karakteristik Basis Data

Menurut Mannino (2001, p4-5), basis data mempunyai beberapa karakteristik, yaitu:

a. Persistent

Sebuah variable dalam program computer tidak akan ada terus menerus. Karena berada dalam memori utama dan akan hilang setelah program dimatikan. Ketika data tidak

diperlukan atau tidak relevan lagi, maka data dapat dihilangkan atau tetap disimpan.

b. Shared

Basis data dapat digunakan oleh siapa saja yang berhak mengakses data tersebut (multiple user). Sebuah basis data menyediakan memori yang secara umum mempunyai banyak fungsi dalam organisasi. Misalnya, basis data dapat mendukung menyediakan laporan-laporan, penyimpanan data kinerja dan lain-lain.

c. Interrelated

Data yang disimpan dalam unit-unit yang terpisah dapat dihubungkan untuk mendapatkan suatu gambaran secara keseluruhan. Misalnya basis data konsumen menghubungkan dengan data pesanan dengan data inventori.

2.2.1.2 Struktur Basis Data

Database memiliki jenjang data yang menyusunnya dari satuan terkecil yaitu karakter sampai menjadi database, dengan susunan :

1. Karakter, merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf, maupun karakter-karakter khusus yang membentuk suatu data atau field.

2. Field, merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data. Kumpulan dari field membentuk suatu record.
3. Record, menggambarkan suatu unit data individu yang tertentu.
4. File, terdiri dari record membentuk suatu file yang menggambarkan satu satuan data yang sejenis.
5. Database, merupakan kumpulan dari file atau table.

2.2.1.3 Komponen Basis Data

Menurut Connolly dan Begg (2002, p18-p20), terdapat 5 komponen utama *database* yaitu sebagai berikut:

1. Hardware: hardware dapat berupa personal computer, mainframe, hingga jaringan komputer. Hardware yang digunakan tergantung pada kebutuhan organisasi dan DBMS yang digunakan.
2. Software : meliputi software database management system (DBMS), sistem operasi, software jaringan dan program aplikasi lain.
3. Data : digunakan oleh organisasi sebagai deskripsi pada data untuk memanggil bagan.

4. Prosedur: instruksi dan aturan yang harus diaplikasikan pada perancangan dan digunakan pada database serta DBMS.
5. Orang : orang-orang yang terlibat didalam sistem.

2.2.2 Database Management System (DBMS)

Menurut Post, Gerald V (2005, p2), *Database Management System* adalah sebuah piranti lunak yang mendefinisikan *database*, penyimpanan data, mendukung *query language*, menghasilkan laporan dan membuat layar memasukan *input data*.

Menurut Connolly dan Begg (2002, p16), DBMS adalah sebuah piranti lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara dan mengontrol akses ke basis data.

Dari definisi-definisi diatas, dapat disimpulkan bahwa DBMS adalah sebuah piranti lunak yang mendukung query language yang dapat memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, mengontrol akses ke basis data.

Menurut Connolly dan Begg (2002, p16) secara khusus DBMS menyediakan fasilitas-fasilitas sebagai berikut:

- Mengijinkan pengguna untuk menentukan basis data, biasanya melalui Data Definition Language (DDL). DDL menyediakan fasilitas untuk menspesifikasikan tipe data, struktur dan batasan data yang bisa disimpan di basis data.

- Mengizinkan pengguna untuk memasukan, mengupdate, menghapus dan mengambil data dari basis data, yang dilakukan melalui *Data Manipulation Language* (DML).
- DBMS juga menyediakan akses kontrol terhadap basis data. Contoh kontrol yang tersedia dalam DBMS:
 - a. *Security system*, yang dapat mencegah pengguna yang tidak terotorisasi mengakses basis data.
 - b. *Integrity system*, menangani konsistensi penyimpanan data.
 - c. *Concurrency and Control system*, yang memungkinkan pembagian akses penggunaan ke basis data.
 - d. *Recovery control system*, yang dapat mengembalikan basis data ke keadaan konsisten awal apabila terjadi kesalahan pada perangkat lunak maupun perangkat keras.

2.2.2.1 Keuntungan dan Kerugian DBMS

Menurut McLeod, Raymond Jr (2004, p193), keuntungan

DBMS antara lain:

1. Mengurangi pengulangan data

Jumlah total file dikurangi dengan menghapus file-file duplikat.

Di dalam DBMS, hanya terdapat sedikit data yang sama di beberapa file.

2. Mencapai independensi data

Spesifikasi data disimpan di dalam skema dari dalam tiap program aplikasi. Perubahan dapat dibuat pada struktur data tanpa mengurangi program yang mengakses data.

3. Mengintegrasikan data dari beberapa *file*

Ketika *file* dibentuk sehingga menyediakan kata *login*, organisasi fisik tidak lagi menjadi kendala.

4. Mengambil data dan informasi secara cepat

Hubungan-hubungan login dan DML (*Data Manipulation Language*) serta query language memungkinkan pemakai mengambil data dalam hitungan detik atau menit, yang sebelumnya mungkin memerlukan beberapa jam atau hari.

5. Meningkatkan keamanan

Baik DBMS, *Mainframe* maupun computer mikro, dapat menyertakan beberapa keamanan seperti *password*, *user directory*, dan *encryption*.

Sedangkan kelemahan DBMS menurut McLeod, Raymond Jr (2004, p152) adalah:

1. Menggunakan *software* yang mahal

DBMS mainframe sangat mahal. DBMS berbasis computer mikro, walaupun biayanya hanya beberapa ratus dollar, dapat menggambarkan pengeluaran yang besar bagi organisasi yang kecil.

2. Memerlukan konfigurasi *hardware* yang besar

DBMS memerlukan kapasitas penyimpanan primer dan sekunder yang lebih besar daripada yang diperlukan oleh program aplikasi lain.

3. Membutuhkan *Database Administrator* (DBA)

DBMS memerlukan pengetahuan khusus agar dapat dimanfaatkan secara maksimal. Pengetahuan khusus ini disediakan paling baik oleh para pengelola basis data (DBA)

2.2.2.2 Fungsi DBMS

Menurut Connolly dan Begg (2002, p48-p52), fungsi DBMS adalah sebagai berikut ini:

1. Sebagai Penyimpanan, Pengambilan, dan Peng-*update*-an data

Sebuah DBMS harus menyediakan user sebuah kemampuan untuk menyimpan, mengambil, dan meng-*update* data dalam DBMS. Ini merupakan fungsi dasar dari DBMS. Dalam menyediakan fungsi ini DBMS harus menyembunyikan detail implementasi fisik internal seperti organisasi file dan struktur penyimpanan dari pengguna.

2. Katalog *User-Accessible*

Sebuah DBMS harus menyediakan sebuah catalog yang menyimpan deskripsi tentang item data dan mudah diakses pengguna.

3. Mendukung Transaksi

Sebuah DBMS harus menyediakan mekanisme yang akan memastikan bahwa semua kegiatan peng-*update*-an yang dilakukan sesuai dengan transaksi yang diberikan atau tidak ada kegiatan *update* yang dibuat bagi transaksi tersebut. Transaksi merupakan sederetan tindakan, yang dilakukan oleh pengguna tunggal atau program aplikasi yang mengakses atau mengubah isi *database*.

4. Layanan Kendali Konkurensi

Sebuah DBMS harus menyediakan sebuah mekanisme untuk memastikan bahwa database di update dengan benar ketika banyak pengguna meng-*update* database secara bersama-sama. Akses bersama relative mudah jika semua pengguna hanya membaca data, dimana tidak ada cara bahwa mereka dapat mengganggu satu dan yang lain. Namun ketika dua atau lebih pengguna mengakses database secara serentak atau bersama-sama dan paling sedikit satu dari mereka mengupdate data, maka disana akan terjadi gangguan yang menghasilkan ketidakkonsistenan.

5. Layanan Perbaikan

Sebuah DBMS harus menyediakan sebuah mekanisme untuk memperbaiki database di saat database mengalami kerusakan dalam berbagai cara. Kerusakan database dapat diakibatkan karena kerusakan system, kesalahan media, dan kesalahan software

atau hardware atau bisa disebabkan karena adanya kesalahan selama proses transaksi dan penyelesaian transaksi yang tidak lengkap.

6. Layanan Authorisasi

Sebuah DBMS harus menyediakan sebuah mekanisme untuk memastikan bahwa hanya pengguna yang berotoritas yang dapat mengakses database. Hal ini untuk mencegah data yang tersimpan tak terlihat oleh semua pengguna dan melindungi database dari akses yang tak berotoritas.

7. Mendukung Komunikasi Data

Sebuah DBMS harus mampu mengintegrasikan dengan software komunikasi. Kebanyakan pengguna mengakses database dari workstation. Kadang workstation tersebut terhubung secara langsung ke computer DBMS. Dalam kasus yang lain, workstation berada pada lokasi yang jauh dan berkomunikasi dengan computer DBMS melalui sebuah jaringan. Dalam hal ini DBMS menerima permintaan sebagai pesan komunikasi dan menanggapi dengan cara yang sama. Semua pengiriman ini ditangani oleh *Data Communication Manager*.

8. Layanan Integritas

Sebuah DBMS harus menyediakan sebuah arti untuk memastikan bahwa data di dalam database dan perubahan pada data mengikuti aturan tertentu. Integritas database dapat mengacu pada

kebenaran dan konsistensi data yang disimpan. Integritas berhubungan dengan kualitas data yang disimpan. Integritas biasanya diekspresikan dengan istilah batasan yaitu berupa aturan konsisten yang tidak boleh dilanggar oleh database.

9. Layanan Peningkatan Keterbebasan Data

Sebuah DBMS harus memasukan sebuah fasilitas untuk mendukung keterbebasan program dari struktur database yang sebenarnya. Data independence biasanya lebih mudah untuk dicapai karena terdapat beberapa jenis perubahan yang dapat dibuat untuk karakteristik fisik dari database tanpa mempengaruhi view. Bagaimanapun data independence loogikal yang lengkap lebih susah untuk di capai.

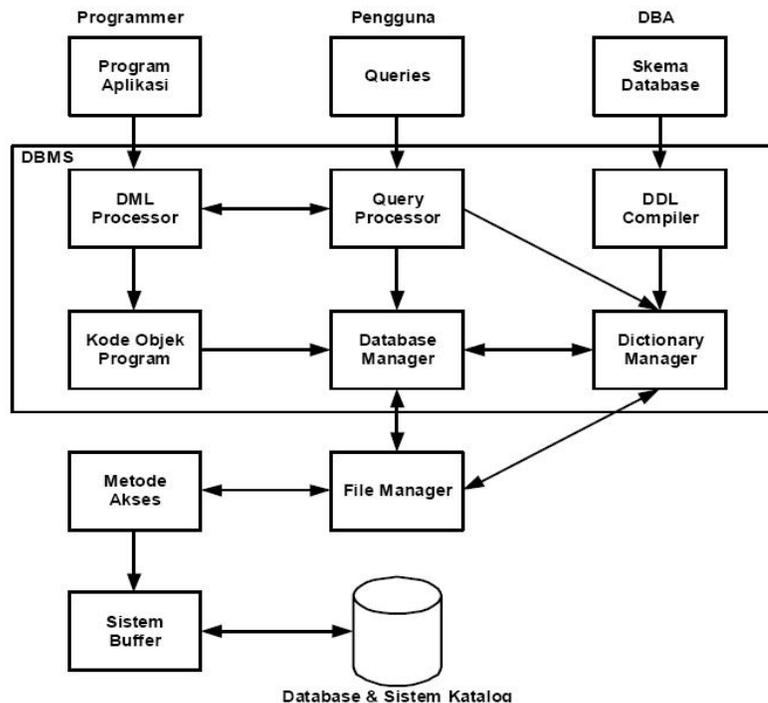
10. Layanan Utilitas

Program Utilitas membantu DBA mengelola database secara efektif. Beberapa utilitas bekerja pada tingkat eksternal, dan konsekuensinya dapat dibuat oleh DBA, yang lainnya bekerja pada tingkat internal dan dapat disediakan hanya dengan vendor DBMS. Contoh dari utilitas tersebut antara lain:

- a. Fasilitas import, untuk load *database* dari flat file, dan fasilitas ekspor, untuk meng- unload *database* pada flat file
- b. Fasilitas Pemantauan, untuk memantau penggunaa dan operasi *database*

- c. Program analisa statistic, untuk memeriksa performa dan penggunaa statistik
- d. Fasilitas penyusunan index, untuk menyusun kembali index dan overflow mereka
- e. Penempatan pengumpulan sampah, untuk menghilangkan record yang dihapus secara fisik dari alat penyimpanan, untuk menggabungkan ruan yang terlepas, dan untuk menempatkan kembali record tersebut dimana ia dibutuhkan.

2.2.2.3 Komponen DBMS



Gambar 2.1 Komponen DBMS
(Connolly & Begg, 2002, p53)

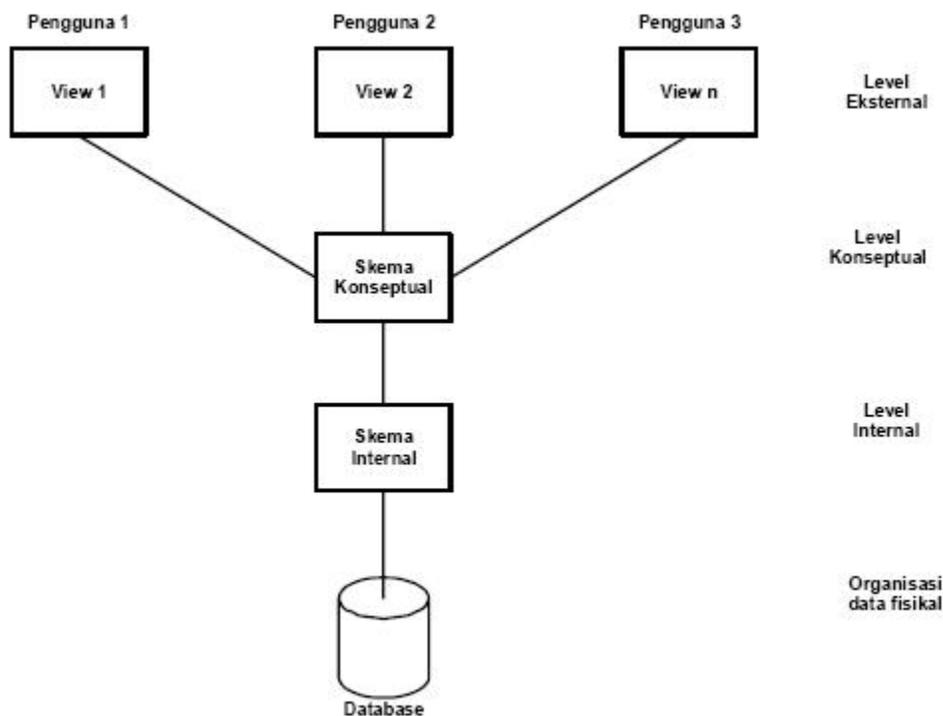
1. *Query Processor* : Merupakan Komponen DBMS yang utama ayng mengubah query ke dalam seperangkat instruksi tingkat rendah langsung ke *database manager*.
2. *Database Manager* : DM menjadi interface degnan program aplikasi user-submitted dan query. DM menerima query dan memeriksa skema eksternal dan konseptual untuk menentukan record konseptual apa yang diperlukan untuk memuaskan permintaan.
3. *File Manager* : File manager memanipulasi penyimpanan file dan mengatur penempatan ruang penyimpanan dalam disk. Komponen ini mendirikan dan memelihara daftar struktur dan index yang didefinisikan dalam skema internal

4. *DML Processor* : Modul ini mengubah pernyataan DML yang tertanam dalam program aplikasi ke dalam panggilan fungsi standard dalam host language. Komponen ini harus berinteraksi dengan query processor untuk membuat kode yang sesuai.
5. *DDL Copiler* : Modul ini mengubah pernyataan DDL kedalam seperangkat tabel berisi meta data. Tabel ini kemudian disimpan dalam catalog sistem sementara itu informasi kendali disimpan dalam header file data.
6. *Catalog Manager* : Catalog manager mengatur akses ke dan memelihara catalog sistem. Katalog sistem diakses oleh sebagian besar komponen DBMS.

2.2.3 Arsitektur ANSI-SPARC Three Level

Menurut Connoly dan Begg (2002, p35-36), arsitektur ANSI-SPARC

Three Level dibagi menjadi tiga bagian yaitu:



Gambar 2.2 ANSI-SPARC Three Level

- Level Eksternal

Eksternal level merupakan level pengguna individu, dimana masing-masing pengguna hanya akan berkepentingan dengan satu bagian saja. Cara pandang dari masing-masing pengguna bersifat abstrak bila dibandingkan dengan bagaimana sebenarnya data tersebut disimpan. Masing-masing pandangan pengguna tersebut disebut external view, yang berisi berbagai tipe eksternal record. Jadi level ini berkaitan erat dengan pemakai, dimana setiap pemakai hanya memerlukan sebagian dari data yang

ada dalam database. Cara pandang secara eksternal hanya terbatas pada entitas, atribut, dan hubungan antar entitas yang diperlukan.

- **Level Konseptual**

Conceptual View merupakan representasi informasi keseluruhan dari isi database, dimana semua pandangan masing-masing pengguna digabungkan. Perwujudan abstrak, bila dibandingkan dengan bagaimana data sesungguhnya tersimpan secara fisik. Conceptual view berisi berbagai tipe record konseptual yang didefinisikan oleh skema konseptual, ditulis dalam data definition language (DDL). Pendefinisian skema konseptual dimaksudkan untuk menyertakan fitur-fitur tambahan, seperti keamanan dan integritas. Beberapa tujuan utama dari skema konseptual diantaranya : menggambarkan perusahaan secara lengkap, bagaimana data tersebut digunakan, bagaimana aliran data di dalam perusahaan, kegunaan data untuk setiap proses, proses kendali atau audit yang diberikan pada setiap proses.

- **Level Internal**

Internal view merupakan level terendah dalam merepresentasikan dari keseluruhan database. Internal view berisikan berbagai tipe internal record yang didefinisikan oleh skema internal. Selain itu juga menyelesaikan mengenai penempatan ruang penyimpanan data dan index, bagaimana perwujudan field-field yang disimpan, deskripsi record untuk

penyimpanan (dengan ukuran penyimpanan untuk elemen), dan teknik enkripsi. Dengan kata lain level ini berkaitan dengan struktur penyimpanan atau database yang tersimpan yang menerangkan tempat penyimpanan data pada internal view, dan denfinisi struktur penyimpanan pada skema internal yang menerangkan hubungannya dengan cara pengaksesan data yang disimpan.

Tujuan dari arsitektur three-level adala untuk menyediakan data independence yang berarti bahwa level yang lebih tinggi tidak terpengaruh oleh perubahan pada level yang lebih rendah.

2.2.4 Pengertian Sistem Basis Data

Menurut Date, C.J (2000, p5), suatu sistem basis data adalah suatu sistem yang pada dasarnya menyimpan banyak record di dalam suatu sistem yang dilakukan secara komputerisasi yang tujuannya secara keseluruhan untuk memelihara informasi dan untuk membuat informasi tersebut tersedia berdasarkan permintaan.

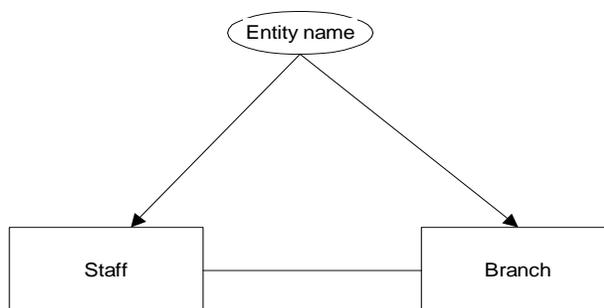
2.2.5 Pemodelan Entity Relationship

Menurut Connolly dan Begg (2002, p330), pemodelan ER adalah pendekatan dari atas ke bawah untuk merancang *database* yang dimulai dengan mengidentifikasi data yang penting yang dikenal dengan sebutan entitas dan hubungan antara data harus diperlihatkan dalam model ini.

Konsep dasar dari pemodelan ER antara lain :

a. Entitas (*Entity*)

Menurut Connolly dan Begg (2002, p331), *entity type* adalah sekumpulan objek dengan properti yang sama, dimana diidentifikasi oleh perusahaan karena mempunyai keadaan bebas. Menurut Connolly dan Begg (2002, p333), *entity occurrence* adalah objek yang didefinisikan secara unik dari *entity type*. *Entity type* digambarkan dalam bentuk bujur sangkar dengan diberi nama entitas, yang umumnya adalah kata benda tunggal, dapat dilihat pada Gambar 2.1.



Gambar 2.3 Diagram dari *Entity Type Branch* dan *Staff*

(Connolly dan Begg, 2002, p333)

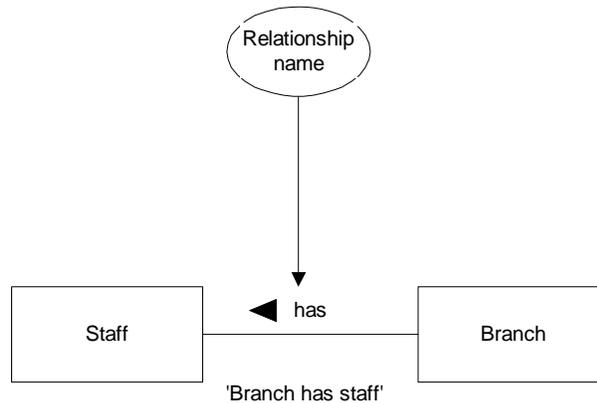
b. Hubungan (*Relationship*)

Menurut Connolly dan Begg (2002, p334), *relationship type* adalah sekumpulan asosiasi berarti antara *entity types*.

Menurut Connolly dan Begg (2002, p334), *relationship occurrence* adalah sebuah asosiasi yang dapat diidentifikasi secara unik dimana terdiri dari satu *occurrence* dari masing-masing *entity type* yang berhubungan.

Setiap *relationship type* digambarkan dengan garis yang menghubungkan *entity type*, dan diberi nama hubungannya seperti pada Gambar 2.2. Pada umumnya, hubungannya menggunakan kata kerja atau frase pendek yang mengandung kata

kerja. Jika memungkinkan, nama hubungannya harus bersifat unik. Secara umum, nama hubungan hanya mempunyai arti untuk satu arah saja. Contohnya : pada gambar di bawah ini menyatakan bahwa *Branch* memiliki *Staff*.



Gambar 2.4 Diagram dari *Relationship Type Branch Mempunyai Staff*

(Connolly dan Begg, 2002, p335)

- Recursive Relationship

Adalah tipe hubungan dimana satu entitas dapat memiliki lebih dari satu peran yang berbeda (*Roles*) contohnya: terdapat suatu entitas *Staff* dimana supervisor juga merupakan *staff* sehingga pada entitas *staff* dapat terjadi hubungan *Staff (Supervisor) supervises Staff(Staff bawahannya)*.

c. Atribut (*Attribute*)

Menurut Connolly dan Begg (2002, p338), atribut adalah properti dari sebuah entitas atau *relationship type*. Contohnya, *Staff* memiliki atribut *staffNo*, nama, posisi, dan gaji.

- Atribut simple dan composite

Simple attribute adalah sebuah atribut yang tersusun dari sebuah *component* dengan keberadaan yang independen. Lalu Composite atribut adalah sebuah atribut yang tersusun dari *multiple components* dengan keberadaan yang independen.

- Atribut *Single-Valued* dan *Multi-Valued*

Single valued attribute adalah sebuah atribut yang memiliki single value pada setiap keberadaannya pada sebuah entitas. Sedangkan *Multi valued attribute* adalah sebuah atribut yang memiliki *multiple values* pada setiap keberadaannya pada suatu entitas.

- *Derived attribute*

Adalah attribute yang mewakili nilai dari atribut yang diwarisi dari atribut lainnya.

- *Keys*

Terdiri dari 3 yaitu: Candidate key, Primary key dan Composite key. *Candidate key* adalah minimal set dari atribut yang secara unik mengidentifikasi setiap keberadaan dari tipe entiy. *Primary key* adalah candidate key yang terpilih sebagai key unik untuk mengidentifikasi suatu keberadaan dari tipe entity. *Composite key* adalah candidate kye yang memiliki dua atau lebih atribut.

- *Strong and weak Entity types*

Strong entity adlah tipe entity yang tidak bergantung pada keberadaan entitas lain. Sedangkan weak entity adalah sebaliknya.

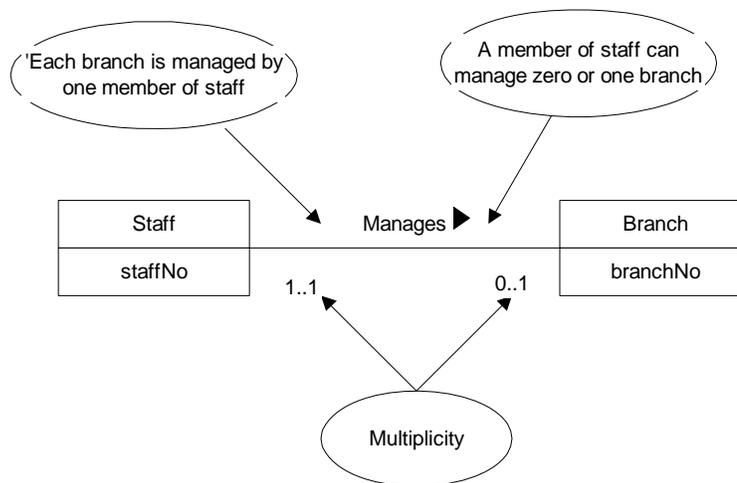
2.2.6 Multiplicity

Menurut Connolly dan Begg (2002, p344), *multiplicity* adalah sejumlah kemunculan yang mungkin ada dalam sebuah entitas yang berhubungan dengan kemunculan tunggal dari entitas yang berhubungan dengannya.

Tiga tipe hubungan *multiplicity* adalah :

a. 1 : 1 (*One-to-One*)

Contoh dari hubungan ini adalah :



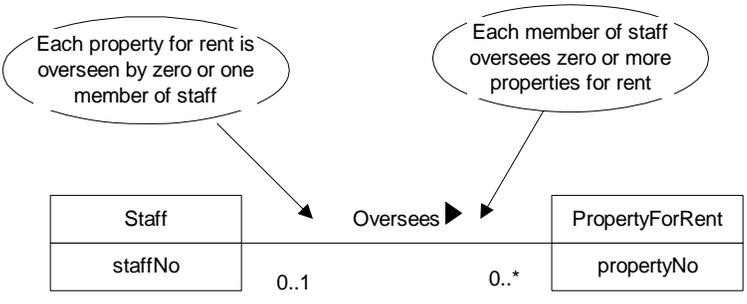
Gambar 2.5 *Multiplicity* Hubungan Satu ke Satu dari *Staff* yang Mengatur *Branch*

(Connolly dan Begg, 2002, p346)

Artinya, seorang staf dapat mengatur nol atau satu cabang dan masing-masing cabang diatur oleh satu staf.

b. 1 : * (*One-to-Many*)

Contoh dari hubungan ini adalah :



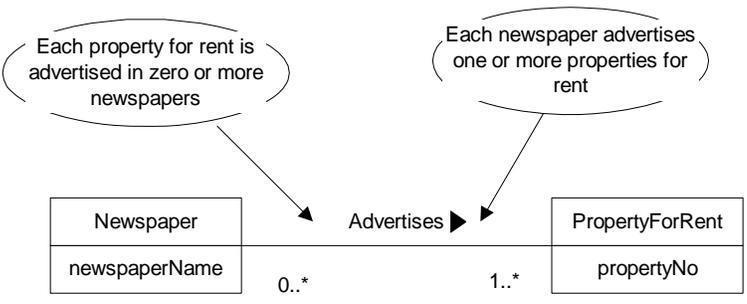
Gambar 2.6 *Multiplicity* Hubungan Satu ke Banyak dari *Staff* yang Mengawasi

PropertyForRent (Connolly dan Begg, 2002, p347)

Artinya, seorang staf dapat mengawasi nol atau lebih properti yang akan disewa dan sebuah properti yang akan disewa diawasi oleh nol atau satu staf.

c. * : * (*Many-to-Many*)

Contoh dari hubungan ini adalah :



Gambar 2.7 *Multiplicity* Hubungan *Many to many* dari *Newspaper* yang Mengiklankan

PropertyForRent (Connolly dan Begg, 2002, p348)

Artinya, satu koran mengiklankan satu atau lebih properti yang akan disewa dan satu properti yang akan disewa diiklankan oleh nol atau lebih koran.

2.2.7 Struktur Data Relational

Menurut Connolly dan Begg (2002, p72-74), Struktur data relasional adalah sebagai berikut:

1. *Relation* : adalah tabel dengan kolom dan baris
2. *Attribute* : adalah nama dari sebuah kolom relasi
3. *Domain* : batasan nilai yang diperbolehkan untuk sebuah atau lebih atribut
4. *Tuple* : adalah baris dari sebuah relasi
5. *Degree* : sebuah degree relasi adalah urutan nomor atribut yang ada
6. *Cardinality* : adalah urutan nomor *tuple* yang ada
7. *Relational Database* : sebuah koleksi dari hubungan yang sudah di normalisasi dengan nama relasi yang berbeda

2.2.8 Relational Keys

Menurut Connolly dan Begg (2002, p78-79), *Relational keys* terdiri dari 4 yaitu sebagai berikut:

1. SuperKey : adalah sebuah atribut atau sekumpulan atribut yang mengidentifikasi secara unik tuple dari sebuah relasi.
2. Candidate key : adalah superkey yang ada dalam suatu relasi
3. Primary Key: adalah candidate key yang paling unik untuk kemudian dipilih untuk mengidentifikasi suatu tuple dalam suatu relasi.
4. Foreign Key: adalah atribut atau sekumpulan atribut dalam suatu relasi yang sama dengan candidate key dari relasi yang lain (atau bisa juga pada relasi yang sama)

2.2.9 Relational Integrity

Menurut Connolly dan Begg (2002, p81-83), *Relational Integrity* terdiri dari 4 yaitu seperti berikut ini:

1. Nulls : Mewaliki nilai dari sebuah atribut yang pada saat tertentu tidak diketahui atau tidak bisa diterapkan pada suatu tuple.
2. Entity Integrity : mengartikan bahwa tidak boleh ada null pada atribut primary key.
3. Referential Integrity : jika sebuah foreign key itu ada maka nilai dari foreign key itu harus cocok dengan candidate key pada tuple yang di refer atau bila tidak haruslah null.
4. Enterprise Constraints : adalah peraturan tambahan yang ditambahkan kemudian oleh user, database administrator

2.2.10 Data Definition Language (DDL)

Menurut Connolly dan Begg (2002, p40) DDL adalah bahasa yang memungkinkan seorang *Database Administrator* atau user untuk mendeskripsikan dan menentukan *entity*, *attribute*, dan relasi yang dibutuhkan aplikasi, bersama dengan semua batasan – batasan integritas dan keamanan (*integrity and security constraints*). Hasil dari kompilasi statement DDL adalah kumpulan dari tabel yang disimpan di file khusus yang disebut *system catalog*. Sementara istilah data *dictionary* dan *data directory* digunakan untuk mendeskripsikan *system catalog*.

2.2.11 Data Manipulation Language (DML)

Menurut Connolly dan Begg (2002, p41) DML adalah bahasa yang menyediakan sekumpulan operasi untuk mendukung operasi manipulasi dasar dari data yang tersimpan di database. Secara umum, operasi DML mencakup:

- Pemasukan data baru ke dalam database (*Insert*)
- Pemoifikasian dari data yang telah tersimpan dalam basis data (*Update*)
- Pengambilan data yang terdapat dalam database (*Select*)
- Penghapusan data yang terdapat dalam basis data (*delete*)

2.2.12 Normalisasi

Normalisasi adalah teknik untuk memproduksi sekumpulan relasi dengan *property* yang diinginkan, dengan kebutuhan data yang diberikan perusahaan. Proses pembentukan tabel normalisasi bertujuan untuk:

- Membuat sekecil mungkin terjadinya *redundancy data*.
- Menghindarkan adanya data yang tidak konsisten terutama bila dilakukan penghapusan atau penambahan data sebagai akibat adanya data rangkap.

Empat bentuk normal yang biasa digunakan, terdiri dari:

a. Bentuk Normal Pertama (*First Normal Form/ 1NF*)

Aturan bentuk normal pertama menurut Connolly dan Begg (2002, p288) adalah sebuah relasi dimana setiap irisan antara baris dan kolom berisikan satu dan hanya satu nilai.

b. Bentuk Normal Kedua (*Second Normal Form/ 2NF*)

Yaitu relasi yang terdapat di dalam 1NF dan tiap atribut *non-primary key*

Bersifat bergantung fungsional penuh terhadap *primary key*.

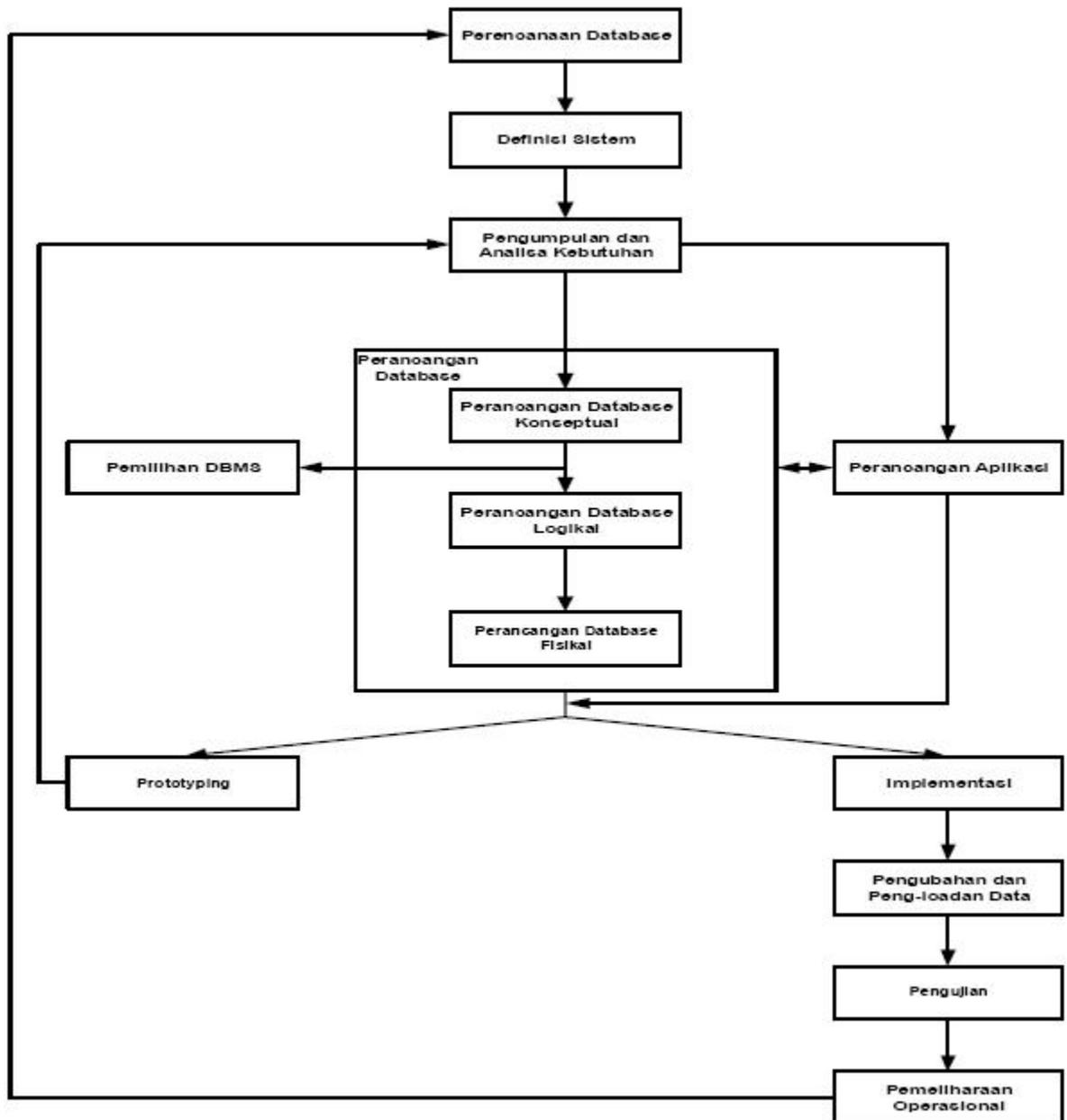
c. Bentuk Normal Ketiga (*Third Normal Form/ 3NF*)

Yaitu relasi yang terdapat pada 1NF dan 2NF, dimana tidak ada atribut *non-primary key* yang bergantung transitif terhadap *primary key*.

d. Bentuk Normal *Boyce-Codd Normal Form/ BCNF*)

Sebuah relasi dikatakan BCNF jika dan hanya jika setiap determinan merupakan *candidate key*. Setiap relasi dalam BCNF juga merupakan 3NF, tetapi relasi dalam 3NF belum tentu termasuk ke dalam BCNF.

2.2.13 Siklus Hidup Aplikasi Database



Gambar 2.8 Siklus Hidup Aplikasi Database
(Connolly dan Begg, 2002, p272)

2.2.13.1 Perencanaan Database

Menurut Connolly (2002, p273), Perencanaan database adalah sebuah aktifitas pengaturan yang memungkinkan langka-langkah dari aplikasi database direalisasikan seefektif dan efisien mungkin.

Hal pertama yang dilakukan adalah mendefinisikan pernyataan sistem bagi proyek database. Pernyataan misi mendefinisikan tujuan utama dari aplikasi database. Pernyataan misi membantu mengklarifikasi tujuan dari proyek database dan menyediakan jalur yang lebih jelas terhadap pembuatan aplikasi database yang efisien dan efektif.

2.2.13.2 Definisi Sistem

Menurut Connolly dan Begg (2002, p274), definisi sistem adalah suatu kegiatan yang menjelaskan bidang dan batasan dari aplikasi database dan user view utama. Batasan dan bidang sistem yang kita buat harus ditentukan tidak hanya untuk pengguna dan area aplikasi saat ini, namun juga untuk pengguna dan area aplikasi di masa depan.

2.2.13.3 Pengumpulan dan Analisa Kebutuhan

Menurut Connolly dan Begg (2002, 276), pengumpulan dan analisa kebutuhan merupakan sebuah proses pengumpulan dan penganalisaan informasi tentang bagian organisasi yang harus didukung oleh aplikasi database, dan menggunakan informasi ini untuk mengidentifikasi kebutuhan pengguna bagi sistem yang baru. Informasi yang didapatkan dari setiap user view utama (peran kerja atau area aplikasi perusahaan) adalah

- Deskripsi data yang digunakan atau dibuat
- Penjelasan bagaimana data dapat digunakan atau di buat
- Berbagai kebutuhan tambahan bagi aplikasi database yang baru

2.2.13.4 Perancangan Database

Menurut Connolly dan Begg (2002, p279), perancangan database merupakan sebuah proses pembuatan sebuah rancangan bagi database yang akan mendukung operasi dan tujuan perusahaan. Tahap-tahap perancangan database dibagi menjadi tiga bagian sebagai berikut:

1. Perancangan Database Konseptual

Menurut Connolly dan Begg (2002, p419), perancangan database konseptual adalah sebuah proses pembuatan model dari informasi yang digunakan dalam perusahaan, yang terbebas dari semua pertimbangan fisik seperti DBMS target, program aplikasi, bahasa pemrograman, hardware dan sebagainya.

2. Perancangan Database Logikal

Menurut Connolly dan Begg (2002, p441), perancangan database logikal merupakan sebuah proses pembuatan model dari informasi yang digunakan dalam sebuah perusahaan berdasarkan pada model data tertentu, tetapi terbebas dari DBMS tertentu dan pertimbangan fisik lainnya.

3. Perancangan Database Fisikal

Menurut Connolly dan Begg (2002, p478), perancangan database fisikal adalah sebuah proses pembuatan deskripsi

implementasi database pada tempat penyimpanan kedua. Perancangan ini menjelaskan relasi dasar, organisasi file, dan index yang digunakan untuk mencapai akses yang efisien ke data dan berbagai batasan integritas yang berhubungan serta penilaian keamanan

2.2.14 Analisa Sistem dan Perancangan Sistem

2.2.14.1 Analisa Sistem

Analisa sistem merupakan proses pengumpulan data dan menaksirkan fakta, pendiagnosaaan masalah, dan menggunakan informasi untuk mengusulkan perbaikan sistem. Suatu sistem akan selalu dianalisa lalu dikembangkan, karena sebuah sistem tidak ada yang sempurna dan selalu terbuka peluang untuk mengembangkannya.

Analisis biasanya dimulai dengan meninjau kembali struktur organisasi dan Uraian kerja personel yang bersangkutan. Disusul dengan formulir, catatan, prosedur, metode, pemrosesan dan laporan yang digunakan oleh perusahaan, termasuk instruksi tertentu kepada personel yang bersangkutan dari rincian prosedur yang terinci, sangat berharga bagi analisis sistem untuk mencari fakta. Langkah-langkah dasar yang harus dilakukan oleh Analisis Sistem sebagai berikut:

1. *Identify*, mengidentifikasi masalah meliputi :
 - a) Mengidentifikasi penyebab masalah
 - b) Mengidentifikasi titik keputusan
 - c) Mengidentifikasi personal – personal dan kunci

2. *Understand*, memahami sistem kerja yang ada, meliputi :
 - a) Menentukan jenis penelitian
 - b) Merencanakan jadwal penelitian
 - c) Membuat agenda pengumpulan data
 - d) Mengumpulkan data
3. *Analyze*, menganalisa sistem meliputi :
 - a) Menganalisa kebutuhan informasi pemakai
 - b) Menganalisa kelemahan sistem yang ada saat ini
 - c) Memberikan alternatif pemecahan masalah (solusi)
4. *Report*, yaitu dengan membuat laporan hasil analisis yang telah dilakukan

2.2.14.2 Perancangan Sistem

Perancangan sistem adalah kombinasi / seri dari proses yang menyangkut aktifitas Identifikasi, Analisa, Menyelesaikan suatu permasalahan.

Dengan kata lain, perancangan sistem adalah proses pengamatan terhadap keadaan suatu badan usaha dengan tujuan dapat mengetahui situasi operasinya dan apakah badan usaha tersebut memerlukan suatu perbaikan atau tidak.

Tujuan perancangan sistem terdiri dari 2 (dua) tujuan, yaitu :

- 1) Untuk memenuhi kebutuhan pemakai sistem
- 2) Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap

kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat. Untuk mencapai tujuan yang kedua, yang lebih condong kepada desain sistem yang terinci, yaitu pembuatan rancangan bangun yang jelas dan lengkap untuk digunakan dalam pembuatan program komputernya. Maka analisis sistem harus dapat mencapai sasaran-sasaran sebagai berikut :

- 1) Perancangan sistem harus berguna, mudah dipahami dan mudah dipergunakan.
- 2) Perancangan sistem harus dapat mendukung tujuan utama perusahaan atau yayasan dengan lebih mendefinisikan pada tahap perencanaan sistem yang selanjutnya adalah tahap analisa sistem.
- 3) Perancangan sistem harus efisien dan efektif untuk dapat mendukung pengolahan transaksi, pelaporan manajemen dan mendukung keputusan yang akan dilakukan oleh komputer.
- 4) Perancangan sistem harus dapat mempersiapkan bangunan yang terinci untuk masing-masing komponen dari sistem informasi.

2.2.15 Perancangan Berorientasi Object

Perancangan berorientasi objek merupakan proses spesifikasi yang terperinci atau pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk. Untuk mengembangkan suatu sistem baru dilakukan dengan menguraikan hubungan proses-proses dalam bentuk diagram-diagram.

Perancangan berorientasi objek bertujuan untuk :

- a. Sistematika proses pendesainan.
- b. Menghasilkan pendesainan model program.
- c. Memberikan gambaran pemecahan masalah dengan efektif.

2.2.16 Data Flow Diagram (DFD)

Menurut Whitten (2004, p326), DFD merupakan model proses yang digunakan untuk menggambarkan aliran data melalui sebuah sistem dan tugas atau pengolahan yang dilakukan oleh sistem.

Ada beberapa kumpulan symbol untuk DFD. Penulis memilih menggunakan notasi atau symbol DeMarco dan Yourdon, sebagai berikut:

a. Process

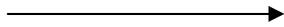
Menyatakan suatu proses atau bagaimana tugas dikerjakan.



Gambar 2.9 Proses

b. Data Flow

Data yang menjadi input atau output ke atau dari suatu proses.



c. Source atau Sink

Orang, unit organisasi, sistem, atau organisasi luar yang berinteraksi dengan sistem. Disebut juga agen eksternal atau entitas eksternal.

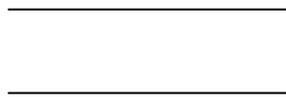


Gambar 2.10 Unit

d. Data Store

Penyimpanan data yang ditujukan untuk penggunaan selanjutnya.

Sering disebut juga dengan file dan database



Gambar 2.11 data store

Didalam DFD terdapat istilah balanced DFD dan unbalanced DFD. balanced DFD berarti banyaknya input untuk level di bawahnya sama dengan banyaknya input untuk level yang lebih tinggi atau banyaknya output untuk level di bawahnya sama dengan banyaknya output untuk level yang lebih tinggi. Jika proses dari context diagram hanya memiliki satu input, sementara input pada level-0 memiliki dua input, berarti terjadi unbalanced DFD.