

BAB 2

LANDASAN TEORI

2.1 Teori-teori Dasar Basis Data

Teori-teori basis data yang akan dibahas pada sub bab ini meliputi : pengertian basis data, tujuan basis data, komponen sistem basis data, database managemen system (DBMS), siklus hidup aplikasi database, pendefinisian sistem (*System Definition*), pengumpulan kebutuhan dan analisis (*Requirement Collection and Analysis*), perancangan basis data (*Database Design*), Perancangan Aplikasi (*Application Design*)emilihan DBMS (*DBMS Selection*), prototipe (*Prototyping*), Implementasi (*Implementation*), perubahan dan pengambilan data (*Data Conversion and Loading*), pengetesan (*Testing*), perawatan operasional (*Operational Maintenance*), tahap-tahap Perancangan basis data, *Entity – Relationship Modelling* (*E-R Modelling*), normalisasi, data *flow diagram* (DFD), *state transition diagram* (STD), Diagram Aliran Dokumen, *Sistem Definition* (Definisi Sistem), *Requirement Collection and Analysis* (Pengumpulan dan Analisis Kebutuhan), Delapan Aturan Emas Perancangan Interface, yang akan dielaborasi pada sub bab-sub bab berikut.

2.1.1 Pengertian Basis Data

Menurut Connolly dan Begg (2005, p15), “Basis data adalah sebuah koleksi logikal data yang saling terhubung satu sama lain dan gambaran dari data tersebut dirancang untuk menemukan kebutuhan informasi pada suatu organisasi/perusahaan”.

Kumpulan data yang berhubungan adalah data yang terintegrasi dengan redundansi atau duplikasi semimum mungkin.

Berhubungan secara logikal maksudnya adalah hubungan data dibuat berdasarkan data model tertentu tanpa memperhatikan pertimbangan fisik. Data model yang digunakan misalnya pemodelan hubungan entitas / *entitas-relationship modeling (ERM)*. Pertimbangan fisik misalnya penggunaan sistem manajemen basis data / *Database Management System (DBMS)* tertentu. *ERM* dan *DBMS* akan dibahas lebih lanjut dalam bab ini.

Basis data dapat digunakan bersama-sama oleh banyak departemen dan pengguna sehingga basis data tidak lagi hanya memenuhi kebutuhan departemen, melainkan kebutuhan dari sebuah organisasi. Selain menyimpan data-data operasional milik organisasi, basis data juga menyimpan data tentang data tersebut, yang sering disebut sistem katalog atau kamus data atau *meta-data* yang ditulis dalam bahasa *data definition language (DDL)*. *DDL* akan dibahas lebih lanjut dalam bab ini.

Menurut Whitten et.al. (2004, p23) data adalah fakta mentah mengenai orang, tempat, kejadian, dan hal-hal penting dalam organisasi.

2.1.2 Tujuan Basis Data

Tujuan basis data Anonimusa(2009) (Sumber : student.eepis-its.edu) terdiri dari :

- a. Kemudahan dan kecepatan dalam pengambilan data (*speed*);
- b. Efisiensi ruang penyimpanan (*space*);
- c. Mengurangi / menghilangkan redundansi data;
- d. Keakuratan (*accuracy*);
- e. Pembentukan kode & relasi antar data berdasar aturan / batasan (*constraint*) tipe data, domain data, keunikan data, untuk menekan ketidakakuratan saat entry / penyimpanan data;
- f. Ketersediaan (*availability*);
- g. Pemilahan data yang sifatnya pasif dari database aktif;
- h. Kelengkapan (*completeness*);
- i. Kompleksnya data menyebabkan perubahan struktur database;
- j. Keamanan (*security*);
- k. Memberikan keamanan atas hak akses data;
- l. Kebersamaan pemakaian (*shareability*);
- m. Bersifat multiuser.

2.1.3. Komponen Sistem Basis Data

Lebih lanjut lagi dalam sebuah sistem basis data, secara lengkap akan terdapat komponen-komponen utama sbb:

- a. Perangkat Keras (hardware);
- b. Sistem Operasi (Operating System);
- c. Basis Data (database);
- d. Sistem (Aplikasi/Perangkat Lunak) Pengelola Bisnis Data (DBMS);
- e. Pemakai (user);
- f. Aplikasi (perangkat lunak) lain (bersifat opsional).

2.1.4 Database Management System (DBMS)

A. Pengertian DBMS

Menurut Whitten et.al. (2004, p520), *Database management system* (DBMS) adalah perangkat lunak khusus yang digunakan untuk membuat, mengontrol, dan mengelola sebuah database.

Menurut Connoly dan Begg (2005, p16), *Database management Sistem* (DBMS) adalah sebuah sistem *software* yang memperbolehkan para pengguna untuk memberi definisi, membuat, menjaga, dan mengontrol akses ke basis data. Pada umumnya DBMS menyediakan fasilitas-fasilitas berikut ini:

1. Mengizinkan para pemakai untuk menggambarkan database, yang pada umumnya melalui *Data Definition Language* (DDL). DDL mengizinkan para pemakai untuk menetapkan struktur dan jenis data serta ketidakeleluasaan atas data untuk disimpan dalam basis data.

Beberapa statement DDL (Connolly dan Begg, 2005, p 168) :

- a. *Create Table*

Create table berfungsi untuk membuat tabel dengan mengidentifikasi tipe data untuk setiap kolom.

- b. *Alter Table*

Alter table berfungsi untuk menambah atau membuang kolom dan *constrain*.

- c. *Drop Table*

Drop table berfungsi untuk membuang atau menghapus tabel beserta semua data yang terkait di dalamnya.

- d. *Create Index*

Create table berfungsi untuk membuat *index*.

- e. *Drop Index*

Drop table berfungsi untuk membuang atau menghapus *index* yang telah dibuat sebelumnya.

2. Mengizinkan para pemakai untuk memasukkan/menyisipkan, memperbaharui, menghapus, dan mendapatkan kembali data dari database, yang pada umumnya melalui suatu *Data Manipulation Language* (DML). DML menyediakan fasilitas penyelidikan umum untuk data, yang disebut sebagai bahasa *query*. Ketentuan dari sebuah bahasa *query* meringankan masalah dalam sistem berbasis arsip dimana pengguna harus bekerja dengan sebuah set *queries* atau ada sebuah perkembangbiakan dari program, memberikan masalah besar pada manajemen *software*. Bahasa *query* yang paling umum adalah *Structured Query Language*, yang sekarang secara formal maupun *de facto* adalah bahasa standar untuk *relational* DBMS. Menurut Connolly dan Begg (2005, p40), DML juga menyediakan operasi manipulasi data pada umumnya meliputi :
 - a. Penyisipan data baru ke dalam basis data (*insertion*);
 - b. Mengubah atau memodifikasi data yang disimpan di dalam basis data (*modify*);
 - c. Pemanggilan data yang ada dalam basis data (*retrieve*);
 - d. Menghapus data dari basis data (*delete*).

Menurut Connolly dan Begg (2005, p41-42), DML dapat dibedakan menjadi dua tipe yang berbeda :

a. Prosedural DML

Sebuah bahasa mengijinkan pemakai memberitahukan sistem untuk data apa yang diperlukan dan bagaimana untuk mendapatkan kembali data secara tepat. Dengan suatu prosedural DML, pemakai atau lebih *programmer* menetapkan data apa yang diperlukan dan bagaimana cara memperoleh data tersebut.

b. Non-Prosedural DML

Sebuah bahasa yang mengijinkan pemakai untuk status data apa yang diperlukan dibandingkan bagaimana untuk mendapatkan kembali. Non-prosedural DML mengijinkan data yang diperlukan untuk memperbaharui statement.

3. Menyediakan akses yang dikendalikan kepada database.

Sebagai contoh, menyediakan :

- a. Suatu sistem keamanan, yang mencegah para pemakai tidak sah yang mengakses database;
- b. Suatu sistem integritas, yang memelihara konsistensi dari data disimpan;
- c. Suatu *concurrency* sitem kendali, yang mengijinkan akses database bersama;

- d. Suatu sistem pengendali *recovery*, memperbaiki kembali database ke status yang konsisten;
- e. Suatu *user-accessible catalog*, yang berisi uraian data di dalam database.

A. Komponen DBMS

Menurut Anonimusb(2009) (sumber : www.dwiantoro.com) Sebuah DBMS (Database Management System) umumnya memiliki sejumlah komponen fungsional (modul) seperti :

- 1) *File Manager*, yang mengelola ruang dalam disk dan struktur data yang dipakai untuk merepresentasikan informasi yang tersimpan dalam disk;
- 2) *Database Manager*, yang menyediakan interface antara data low-level yang ada di basis data dengan program aplikasi dan query yang diberikan ke sistem. Query Processor, yang menterjemahkan perintah-perintah dalam query language ke perintah low-level yang dapat dimengerti oleh database manager;
- 3) *DML Precompiler*, yang mengkonversi perintah DML yang ditambahkan dalam sebuah program aplikasi ke paman gil prosedur normal dalam bahasa induk;

- 4) *DDL Compiler*, yang mengkonversi perintah-perintah DDL ke dalam sekumpulan tabel yang mengandung metadata. Tabel-tabel ini kemudian disimpan dalam kamus data.

B. Fungsi-fungsi DBMS

- 1) Untuk menyimpan, mendapatkan kembali, dan memperbaharui data;
- 2) Pemakai dapat mengakses data;
- 3) Mendukung transaksi;
- 4) *concurrency control services*, suatu DBMS harus melengkapi suatu mekanisme untuk memastikan bahwa database diperbaharui dengan tepat ketika berbagai para pemakai sedang memperbaharui database secara bersamaan;
- 5) *Recovery service*, suatu DBMS harus melengkapi suatu mekanisme untuk me-covery database database dirusak;
- 6) *Authorization services*, suatu DBMS harus melengkapi suatu mekanisme untuk memastikan bahwa hanya pemakai yang diberi hak dapat mengakses data;
- 7) Pendukung komunikasi data;
- 8) *Integrity service*, suatu DBMS melengkapi makna untuk memastikan bahwa kedua-duanya data di dalam database dan merubah data mengikuti kepada aturan tertentu;

- 9) *Services to promote data independence*, suatu DBMS harus meliputi fasilitas untuk mendukung independen program dari struktur database yang nyata;
- 10) *Utility services*, suatu DBMS perlu menyediakan seperangkat utilitas pelayanan.

C. Keuntungan dan kerugian DBMS

DBMS memiliki keuntungan dan kerugian menurut Connolly dan Begg (2005, p26-30).

Keuntungan dari DBMS, terdiri dari :

- 1) Kontrol terhadap pengulangan data;
- 2) Data yang dihasilkan konsisten;
- 3) Data dapat dipakai secara bersama-sama;
- 4) Meningkatkan integritas data;
- 5) Meningkatkan keamanan;
- 6) Penetapan standarisasi;
- 7) Perbandingan skala ekonomi;
- 8) Mengatasi konflik kebutuhan;
- 9) Memperbaiki pengaksesan data;
- 10) Meningkatkan produktivitas;
- 11) Memperbaiki pemeliharaan data melalui data yang tidak tergantung dengan data yang lain;

12) Memperbaiki pengaksesan data secara bersama-sama;

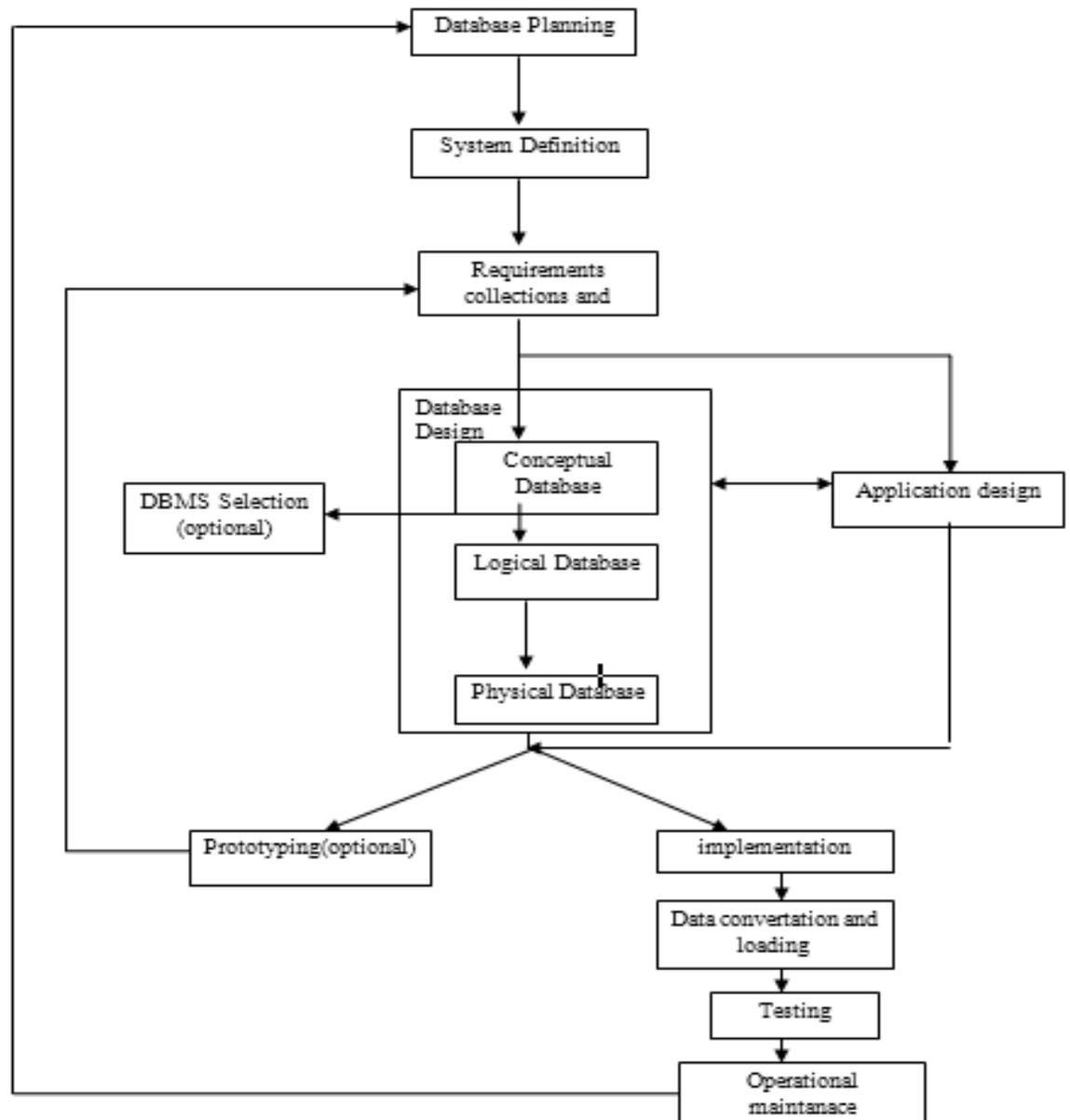
13) Memiliki *back up* data.

Kerugian penggunaan DBMS, terdiri dari :

- 1) Memiliki sistem yang kompleks;
- 2) Karena sistem yang kompleks mengakibatkan DBMS memiliki ukuran yang semakin besar;
- 3) DBMS memiliki harga yang bervariasi tergantung dari fungsi dan kebutuhan;
- 4) Penambah biaya untuk perangkat keras yang dibutuhkan.

2.1.5 Siklus Hidup Aplikasi Database

Menurut Connolly dan Begg (2005, p284), siklus hidup sebuah aplikasi basis data digambarkan sebagai berikut:



(Sumber : Connolly dan Begg, 2005, p284)

Gambar 2.1 Siklus Hidup Aplikasi Basis Data

Menurut Connolly dan Begg (2005 , p285), perencanaan basis data merupakan aktivitas-aktivitas manajemen yang memungkinkan tahap-tahap dalam aplikasi basis data direalisasikan seefisien dan seefektif mungkin. Perencanaan basis data harus diintegrasikan dengan keseluruhan sistem informasi suatu organisasi.

Ada tiga persoalan utama yang terlibat dalam perumusan suatu strategi sistem informasi :

- a. Identifikasi rencana, sasaran (*goals*) dan tujuan perusahaan dengan penentuan kebutuhan sistem informasi;
- b. Evaluasi sistem informasi yang sedang berjalan untuk menentukan kelebihan dan kekurangan yang ada;
- c. Penilaian terhadap peluang IT apakah mampu menghasilkan keuntungan yang kompetitif.

2.1.6 Pendefinisian Sistem (*System Definition*)

Menurut Connolly dan Begg (2005, p286), pendefinisian sistem (*System Definition*) menjelaskan bidang dan batasan aplikasi basis data serta pandangan pengguna. Hal ini sangat penting dilakukan dalam suatu proses perancangan basis data agar dapat melakukan proses identifikasi mengenai batasan sistem yang akan dirancang, serta bagaimana sistem.

2.1.7 Pengumpulan Kebutuhan dan Analisis (*Requirement Collection and Analysis*)

Menurut Connolly dan Begg (2005, p288), pengumpulan dan analisis kebutuhan adalah proses pengumpulan analisis informasi tentang bagian perusahaan yang didukung oleh aplikasi basis data dan yang menggunakan informasi ini untuk mengidentifikasi kebutuhan-kebutuhan user dari sistem yang baru.

2.1.8 Perancangan Basis Data (*Database Design*)

Perancangan Basis data menurut Connolly dan Begg (2005, p291), merupakan proses menciptakan perancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan.

Menurut Connolly dan Begg (2005, p2923), terdapat dua pendekatan yang dapat digunakan untuk merancang basis data, yaitu:

a. Bottom up approach

Bottom up dimulai dari atribut awal (entiti dan relationship) yang dianalisa asosiasi antar atribut, kemudian dibentuk relation yang mewakili tipa dari entiti dan relationship antar entiti.

b. Top down approach

Pendekatan top down dimulai dengan pengembangan data mode yang terdiri dari sedikit atau banyak entity dan relationship, kemudian

melakukan perbaikan top down untuk mengidentifikasi lower-level entity, relationship, dan asosiasi antar atribut.

Menurut Connolly dan Begg (2005, p293), dalam *database design* terdapat tiga fase utama, yaitu perancangan konseptual (Conceptual database design), perancangan logikal (Logical database design), fisik (Physical database design).

- a. Menurut Connolly dan Begg (2005, p293), perancangan basis data konseptual merupakan proses dalam membangun suatu model informasi yang digunakan di dalam perusahaan, dimana proses tersebut tidak tergantung oleh semua pertimbangan fisik;
- b. Perancangan basis data logikal menurut Connolly dan Begg (2005, p294), merupakan proses dalam membangun suatu model informasi yang digunakan di dalam perusahaan, proses tersebut tidak tergantung oleh suatu DBMS dan pertimbangan fisik tertentu, tetapi berdasarkan pada model data spesifik yang diperlukan;
- c. Perancangan basis data fisik menurut Connolly dan Begg (2005, p294), merupakan proses untuk menghasilkan gambaran dari implementasi basis data pada tempat penyimpanan, menjelaskan dasar dari relasi, organisasi file dan index yang digunakan untuk efisiensi data dan menghubungkan beberapa *integrity constraction* dan tindakan keamanan.

2.1.9 Pemilihan DBMS (*DBMS Selection*)

Menurut Connolly dan Begg (2005, p295), pemilihan dari suatu DBMS yang tepat sangatlah perlu untuk mendukung aplikasi basis data yang efektif.

Tahap-tahap pemilihan DBMS :

- a. Mendefinisikan kriteria berdasarkan spesifikasi kebutuhan pemakai;
- b. Menentukan beberapa produk DBMS;
- c. Mengevaluasi produk-produk tersebut;
- d. Merekomendasikan produk DBMS tertentu.

2.1.10 Perancangan Aplikasi (*Application Design*)

Menurut Connolly dan Begg (2005, p299), perancangan aplikasi adalah perancangan antarmuka pengguna dan program aplikasi yang menggunakan dan memroses basis data. Ada dua aspek penting dalam perancangan aplikasi, yaitu :

- a. Perancangan Transaksi (*Transaction Design*)

Transaksi merupakan sebuah aksi, atau serangkaian aksi yang dilakukan oleh seorang pengguna atau program aplikasi yang mengakses atau mengubah isi dari basis data. Tujuan dari perancangan transaksi adalah untuk menetapkan dan mendokumentasikan karakteristik tingkat tinggi.

Perancangan harus dilakukan lebih awal dalam proses perancangan untuk memastikan bahwa basis data yang diimplementasikan mampu mendukung semua transaksi yang dibutuhkan. Ada tiga jenis transaksi, yaitu :

- 1) *Retrieval transactions*, digunakan untuk mendapatkan kembali data untuk ditampilkan di layar atau dalam laporan.
- 2) *Update transaction*, digunakan untuk menambah data, menghapus data lama, atau memodifikasi data yang ada dalam basis data.
- 3) *Mixed transaction*, melibatkan *retrieval* (pemanggilan) dan *update* (perubahan) data atau kombinasi antara keduanya.

b. Perancangan antarmuka (*User Interface Design*)

Sebelum mengimplementasikan suatu *form* atau laporan, ada perlunya merancang tampilan (*layout*) terlebih dahulu. Berikut pedoman yang berguna dalam perancangan laporan.

1. Judul yang berarti, diusahakan pemberian nama suatu *form* cukup jelas menerangkan kegunaan dari suatu *form* atau laporan.
2. Instruksi yang dapat dipahami, menggunakan terminologi yang lazim dalam menyampaikan instruksi kepada pengguna.

Instruksi harus diuraikan dengan singkat dan jelas, dan ketika membutuhkan informasi lebih lanjut, layar bantuan harus tersedia.

3. Pengelompokkan logis dan pengurutan *field*, *field* yang berhubungan harus ditempatkan secara bersama dalam suatu *form*/laporan pengurutan *field* harus logis dan konsisten.
4. Tampilan permohonan *layout form* laporan secara visual, *form*/laporan harus menarik perhatian bagi pengguna.
5. Terminologi dan singkatan yang digunakan harus konsisten.
6. Penggunaan warna secara konsisten dan berarti.
7. Ruang yang tampak dan batasan untuk *field* pemasukan data, seorang pengguna harus secara visual menyadari jumlah ruang yang tersedia untuk setiap *field*.
8. Pergerakan kursor yang baik, seorang pengguna harus dengan mudah mengenal operasi yang dibutuhkan untuk menggerakkan kursor di seluruh *form*/laporan.
9. Perbaiki kesalahan untuk karakter individual dan *field* secara keseluruhan.
10. Pesan kesalahan untuk nilai yang tidak dapat diterima sistem.
11. *Field-field* yang bersifat pilihan harus ditandai dengan jelas.
12. Pesan-pesan untuk *field* yang bersifat menjelaskan.

Ketika suatu pengguna menempatkan kursor pada suatu *field*, informasi tentang *field* tersebut seharusnya muncul dalam suatu posisi yang teratur pada layar.

13. Tanda penyelesaian

Indikator yang menjelaskan bahwa suatu proses telah selesai dilaksanakan. Begitu juga ketika proses pengisian dalam *field* pada suatu *form* lengkap.

2.1.11 Prototipe (*Prototyping*)

Menurut Connolly dan Begg (2005, p303-304), *prototyping* adalah membuat model kerja dari aplikasi basis data. Tujuan utama dari pengembangan *prototype* aplikasi basis data adalah untuk memungkinkan pengguna memakai *prototype* tersebut dalam mengidentifikasi kelebihan dan kekurangan sistem, dan memungkinkan perancang untuk memperbaiki atau melengkapi fitur-fitur aplikasi basis data yang baru.

2.1.12 Implementasi (*Implementation*)

Menurut Connolly dan Begg (2005, p304), adalah realisasi fisik dari basis data dan rancangan aplikasi, membuat eksternal, konseptual, dan definisi basis data dilakukan dengan menggunakan *Data Definition Language (DDL)* dari DBMS yang terpilih. Bagian

dari program aplikasi ini merupakan transaksi-transaksi basis data yang diimplementasikan menggunakan *Data Manipulation Language (DML)*.

2.1.13 Perubahan dan Pengambilan Data (*Data Conversion and Loading*)

Menurut Connolly dan Begg (2005, p305), merupakan pemindahan data yang ada ke dalam basis data baru dan mengubah aplikasi yang ada untuk beroperasi pada basis data yang baru. Langkah ini diperlukan hanya ketika suatu sistem basis data baru menimpa sistem yang lama.

2.1.14 Pengetesan (*Testing*)

Menurut Connolly dan Begg (2005, p305), langkah ini adalah mengadakan testing terhadap aplikasi basis data dengan tujuan untuk melihat apakah masih ada kesalahan dan memvalidasi sesuai dengan keinginan *user*. Pengujian ini dilakukan dengan pengujian yang direncanakan dengan hati-hati berdasarkan data yang realistis.

2.1.15 Perawatan Operasional (*Operational Maintenance*)

Menurut Connolly (2005, p306) merupakan proses pengawasan dan pertahanan sistem berikut instalasi. Pada langkah sebelumnya, aplikasi basis data telah diimplementasikan dan diuji sepenuhnya.

2.1.16 Tahap-tahap Perancangan Basis data

Dalam merancang suatu basis data melalui beberapa tahapan, sebagai berikut :

- a. Perancangan Basis data Konseptual
- b. Perancangan Basis data Logikal
- c. Perancangan Basis data Fisikal

A. Perancangan Basis data konseptual

Menurut Connolly dan Begg (2005, p439), perancangan konseptual basis data adalah proses pembangunan model dan data yang digunakan di perusahaan, yang tidak bergantung pada semua pertimbangan fisikal. Tujuannya untuk membangun representasi konseptual basis data, yang meliputi identifikasi dari entitas-entitas, *relationship – relationship*, dan atribut – atribut yang penting.

Menurut Connolly dan Begg (2005, p443), langkah – langkah perancangan konseptual basis data yaitu :

Langkah 1. Membangun *conceptual data model*

Menurut Connolly dan Begg (2005,p442), tujuannya adalah untuk membangun *local conceptual* dari perusahaan untuk tiap *view* yang spesifik. Langkah – langkah yang harus dilakukan pada langkah 1:

Langkah 1.1 Identifikasikan Tipe Entitas

Menurut Connolly dan Begg (2005, p443), Tujuan dari identifikasi entitas adalah Mengidentifikasi entitas utama yang diperlukan oleh tiap *view*.

Langkah 1.2 Identifikasi Tipe *Relationship*

Menurut Connolly dan Begg (2005, p445), tujuannya adalah mengidentifikasi *relationship-relationship* penting yang ada diantara tipe-tipe entitas yang telah diidentifikasi.

Langkah 1.3 Mengidentifikasi dan Mengasosiasikan Atribut dengan Tipe Entitas dan Tipe *Relationship*

Menurut Connolly dan Begg (2005, p447), tujuannya adalah untuk menghubungkan atribut-atribut dengan entitas atau *relationship type* yang sesuai.

Langkah 1.4 Menentukan atribut domain

Menurut Connolly dan Begg (2005, p450), tujuannya adalah untuk menentukan domain-domain untuk atribut-atribut dalam *local conceptual data model*.

Langkah 1.5 Menentukan *Candidate Key dan Primary Key*

Menurut Connolly dan Begg (2005, p451), tujuannya adalah mengidentifikasi candidate key untuk tiap entitas dan jika ada lebih dari satu candidate key, pilih satu untuk primary key. Sisa dari candidate key dinamakan alternate key.

Langkah 1.6 Mempertimbangkan penggunaan dari konsep *enchaced modelling*

Menurut Connolly dan Begg (2005, p453), Tujuannya adalah mempertimbangkan penggunaan dari konsep enchanced modelling seperti spesialisasi, generalisasi, agregasi atau composition.

Langkah 1.7 Validasi Model Konseptual dengan *User Transaction*

Menurut Connolly dan Begg (2005, p456), tujuannya adalah untuk menjamin bahwa model konseptual mendukung transaksi-transaksi yang dilakukan oleh *view*.

B. Perancangan Basis data Logikal

Perancangan basis data logikal adalah proses membangun sebuah model dari informasi yang digunakan dalam perusahaan

berdasar sebuah model data yang khusus, tetapi lepas dari DBMS khusus dan pertimbangan fisikal lainnya.

Langkah 2 : Membangun dan memvalidasi model data logikal lokal untuk setiap view

Menurut Connolly dan Begg (2005, p462), tujuannya adalah menerjemahkan *logical data model* dan kemudian untuk memvalidasi model tersebut untuk memeriksa model tersebut benar secara struktural dan memiliki kemampuan untuk mendukung transaksi-transaksi yang dibutuhkan. Tujuan ini akan tercapai dengan mengikuti langkah-langkah berikut :

Langkah 2.1 Menentukan Relasi-Relasi untuk Model Data Logikal

Menurut Connolly (2005, p463) tujuannya adalah menciptakan relasi-relasi untuk model data logikal untuk merepresentasikan entitas-entitas, relasi-relasi, dan atribut-atribut yang telah diidentifikasi.

Tujuan dari langkah ini adalah :

1. Menghilangkan tipe hubungan binary many-to-many (*:*)
2. Menghilangkan tipe hubungan rekursif many-to-many (*:*)
3. Menghilangkan tipe hubungan yang kompleks
4. menghilangkan atribut yang multi value.

Langkah 2.2 Menganalisis relasi untuk model data logikal lokal

Tujuannya adalah membuat relasi untuk model data logikal lokal dengan menggambar entitas, hubungan, dan atribut yang telah diidentifikasi.

Cara relasi didapatkan dari struktur yang mungkin ditunjukkan dalam model data :

1. Tipe entitas kuat
2. Tipe entitas lemah
3. Tipe hubungan binary one-to-many (1:*)
4. Tipe hubungan binary one-to-one (1:1)
5. Hubungan rekursif one-to-one (1:1)
6. Tipe hubungan superclass/subclass
7. Tipe hubungan binary many-to-many (*:*)
8. Tipe hubungan rumit
9. Atribut multi nilai

Langkah 2.3 Validasi relasi menggunakan normalisasi

Tujuan dari normalisasi adalah sebagai berikut :

- a. Menghilangkan kumpulan relasi dari ketergantungan yang tidak diharapkan atas kegiatan penambahan, pengeditan dan penghapusan data.

- b. Mengurangi kebutuhan restrukturisasi kumpulan relasi dan meningkatkan file spam program aplikasi.
- c. Membuat model relasional lebih informatif.

Langkah 2.4 Validasi relasi dengan transaksi *user*

Tujuannya adalah meyakinkan bahwa relasi dalam model data logikal lokal mendukung transaksi yang dibutuhkan user.

Jika dapat menjalankan semua transaksi dengan cara manual, kita telah memvalidasi model data logikal lokal dengan transaksi secara manual, maka terdapat masalah dengan model data.

Langkah 2.5 Mengecek integritas basis data

Tujuannya adalah mendefinisikan ruang lingkup integritas yang diperlihatkan kepada user.

Setelah mengidentifikasi ruang lingkup integritas, akan didapatkan model data logikal lokal yang lengkap dan gambaran akurat dari sebuah view.

Terdapat lima tipe dari pembatas integritas :

1. Data yang dibutuhkan;
2. Batasan domain;
3. Integritas entity;

4. Integritas referensi;
5. Pembatas perusahaan.

Langkah 2.6 *Mereview* model data logikal lokal dengan *user*

Tujuannya adalah meyakinkan model data logikal lokal dan membuat dokumentasi yang menjelaskan model tersebut sebagai gambaran yang sesuai dengan keadaan sebenarnya.

C. Perancangan Basis data Fisikal

Menurut Connolly dan Begg (2005, p439), perancangan fisik basis data adalah proses membuat deskripsi dari implementasi basis data pada *secondary storage*, mendeskripsikan relasi dasar, *file organization*, dan *index* yang digunakan untuk mendapatkan akses efisien pada data dan semua *inegrity constraint* yang berhubungan dan *security measure*. Tujuannya adalah untuk memutuskan bagaimana struktur logikal diimplementasikan (sebagai relasi dasar) secara fisik dalam DBMS yang dipilih.

Menurut Connolly dan Begg (2005, p496, langkah-langkah perancangan fisik basis data meliputi :

Langkah 3 : Menerjemahkan logical data model untuk DBMS yang dipilih

Menurut Connolly (2005, p497) Tujuannya adalah untuk menghasilkan *relational database schema* dari data model logikal yang dapat diimplementasikan dalam DBMS yang dipilih. 3 (tiga) aktivitas pada step 3 :

Langkah 3.1 Merancang relasi-relasi dasar

Menurut Connolly dan Begg (2005, p498), tujuan adalah untuk memutuskan bagaimana merepresentasikan relasi-relasi dasar yang diidentifikasi dalam model data logikal dalam DBMS yang dipilih. Untuk tipe relasi yang diidentifikasi dalam model data logikal, kita mempunyai definisi yang terdiri dari :

- Nama relasi
- Daftar atribut-atribut sederhana dalam golongan-golongan
- *Primary key* dan *alternate key* dan *foreign* jika ada
- *Referential integrity constraints* untuk semua *foreign keys* yang diidentifikasi.

Menurut Connolly dan Begg (2005,p498), sedangkan dari *data dictionary*, dari tiap-tiap atribut kita juga mempunyai :

- Domain-nya, yang terdiri dari tipe data, panjang, dan semua batasan dalam domain
- Nilai *default optional* untuk atribut
- Apakah atribut dapat mempunyai nilai *null*
- Apakah atribut tersebut *derived*, maka harus dikomputasi.

Langkah 3.2 Merancang Representasi untuk Data Turunan

Menurut Connolly dan Begg (2005, p499), tujuannya adalah untuk memutuskan bagaimana merepresentasikan semua data *derived* yang ada dalam model data logikal dalam DBMS yang dipilih.

Langkah 3.3 Merancang General Constraint

Menurut Connolly dan Begg (2005, p501), tujuannya adalah untuk merancang *general constraint* untuk DBMS yang dipilih.

Langkah 4 Merancang file organization dan indexes

Menurut Connolly dan Begg (2005, p501), tujuannya adalah untuk menentukan file organisasi yang optimal untuk menyimpan relasi-relasi dasar dan indeks-indeks yang dibutuhkan

untuk mencapai performansi yang dapat diterima, dengan begitu relasi dan *tuple* akan disimpan pada *secondary storage*.

Langkah 4.1 Menganalisa Transaksi

Menurut Connolly dan Begg (2005 , p502), tujuannya adalah untuk mengerti fungsi dan transaksi yang akan diterapkan pada basis data dan untuk menganalisa transaksi-transaksi yang penting.

Langkah 4.2 Memilih Organisasi file

Menurut Connolly dan Begg (2005 , p506), tujuannya adalah untuk menentukan organisasi file yang efisien untuk tiap relasi dasar.

Langkah 4.3 Memilih index – index

Memilih Connolly dan Begg (2005 , p508), tujuannya adalah untuk mempertimbangkan apakah dengan menambahkan index akan meningkatkan performansi dari sistem.

Langkah 4.4 Memperkirakan kebutuhan *disk space*

Menurut Connolly dan Begg (2005 , p514), untuk memperkirakan jumlah *disk space* yang dibutuhkan oleh basis

data. Untuk menyimpan data dan semua *index-index nonclustered* dasar dalam tabel yang tidak mempunyai *clustered index* maka kita dapat menggunakan beberapa langkah berikut :

1. Menghitung *space* yang digunakan untuk menyimpan data
2. Menghitung *space* yang digunakan untuk menyimpan tiap *index-index nonclustered* dasar
3. Menjumlahkan nilai – nilai yang dikalkulasikan

Langkah 5. Merancang *user view*

Menurut Connolly dan Begg (2005 , p515), tujuannya adalah untuk merancang *user view* yang diidentifikasi selama pengumpulan kebutuhan–kebutuhan dan tahap analisis dari *system development lifecycle*.

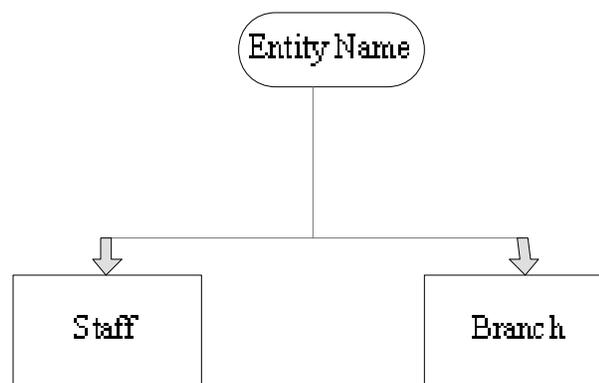
Langkah 6. Merancang mekanisme keamanan

Menurut Connolly dan Begg (2005 , p516), tujuannya adalah untuk merancang mekanisme keamanan untuk basis data seperti yang ditentukan oleh *user* pada waktu tahap pengumpulan kebutuhan–kebutuhan dari *system development lifecycle*. Mekanisme keamanan yang dirancang dalam basis data yaitu mekanisme keamanan sistem dan mekanisme keamanan sistem.

2.1.17 Entity – Relationship Modelling (E-R Modelling)

A. Entity Type

Entity type adalah sekumpulan objek yang memiliki properti yang sama yang diidentifikasi dalam perusahaan serta keberadaannya independen Conolly dan Begg (2005, p343), setiap objek yang diidentifikasi secara unik disebut entity occurrence (Connolly dan Begg, 2005, p333).



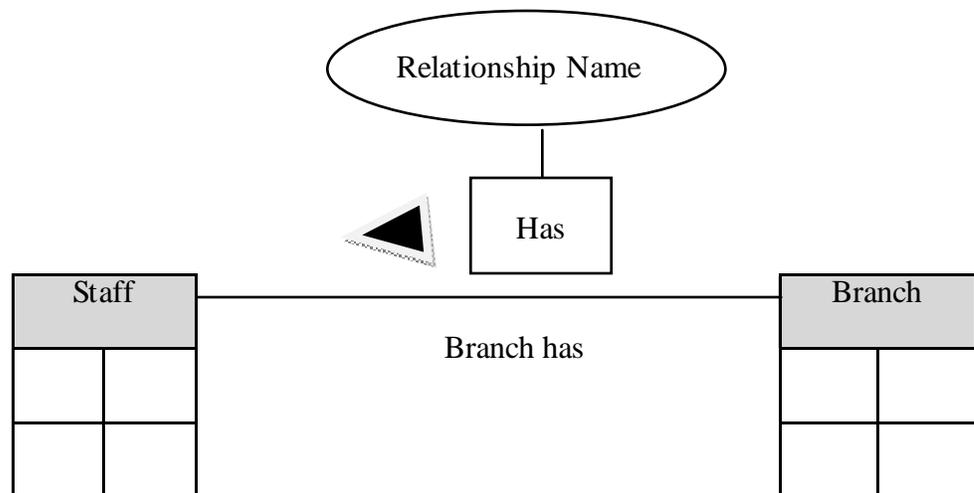
Gambar 2.2 Representasi Diagram Dari *Entity Type Staff* dan *Branch*

Entity Occurance adalah suatu objek dari tipe entitas yang dapat diidentifikasi secara unik. Tipe entitas dibedakan menjadi dua yaitu :

1. *Strong entity type* yaitu tipe entitas yang keberadaannya tidak bergantung kepada tipe entitas lain.
2. *Weak entity type* yaitu tipe entitas yang keberadaannya bergantung kepada tipe entitas yang lain.

B. *Relationship Type*

Relationship Type adalah sekumpulan *entity* yang mempunyai hubungan dan memiliki arti (Connolly dan Begg, 2005, p346).



(Sumber : Connolly, 2005, p347)

Gambar 2.3 Representasi Diagram dari *Entity Branch Has Staff Relationship type*

C. *Atribut*

Menurut Connolly dan Begg (2005, p350), atribut adalah sebuah *property* dari sebuah entitas ada tipe hubungan (*relationship*). Atribut *Domain* adalah sekumpulan nilai yang diperbolehkan untuk satu atau lebih atribut.

D. Keys

Menurut Connolly dan Begg (2005, p352), *keys* dapat dikelompokkan sebagai berikut :

1. *Candidate key*

Sekumpulan kecil atribut yang mengidentifikasi setiap kejadian dari sebuah tipe entitas secara unik.

2. *Primary key*

Candidate key yang dipilih untuk mengidentifikasi setiap kejadian dari tipe entitas secara unik.

3. *Composite key*

Candidate key yang memiliki dua atau lebih atribut.

4. *Foreign key*

Sebuah atribut atau sekumpulan atribut yang memiliki hubungan yang cocok dengan *candidate key* dari beberapa hubungan yang lainnya.

E. Structural Constraints

Menurut Connolly dan Begg (2005, p356), *multiplicity* adalah jumlah dari kejadian yang mungkin dari sebuah tipe entitas yang berhubungan kepada sebuah kejadian tunggal dari tipe entitas yang terasosiasi melalui *relationship* (hubungan) tertentu. Derajat yang

biasa digunakan untuk *relationship* (hubungan) adalah *binary*.

Binary relationship terdiri atas :

1. *One –to-One (1:1) Relationship*
2. *One-to-Many (1:*) Relationship*
3. *Many-to-Many (*:*) Relationship*

Multiplicity juga mengandung dua *constraints* yang terpisah meliputi :

1. *Cardinality*, menjelaskan jumlah maksimum dari *relationship occurrence* yang mungkin untuk sebuah entitas yang berpartisipasi dalam tipe *relationship* (hubungan) yang diberikan.
2. *Participation*, menentukan apakah semua atau hanya beberapa *entity occurrence* yang berpartisipasi dalam *relationship* (hubungan).

2.1.18 Normalisasi

Normalisasi menurut Connolly dan Begg (2005, p388), adalah suatu teknik untuk menghasilkan sebuah relasi dengan properti-properti yang diinginkan, memberikan kebutuhan data dari sebuah perusahaan.

Tujuan normalisasi :

1. Terjaminnya struktur yang konsisten
2. Kerangkaan yang minimal

3. Stabilitas struktur data yang maksimal

Manfaat normalisasi :

1. Meminimalkan jumlah kapasitas penyimpanan yang diperlukan untuk menyimpan data.
2. Meminimalkan resiko data yang tidak konsisten dalam suatu basis data.
3. Meminimalkan kemungkinan *update* dan *delete anomaly*
4. Memaksimalkan stabilitasi dan struktur data

Adapun beberapa tahapan dari normalisasi :

1. *Normalisasi pertama (1NF)*

Menurut Connolly dan Begg (2005, p403), bentuk normal pertama (1NF) adalah relasi dimana pertemuan antar setiap baris dan kolom terdiri satu dan hanya satu nilai. Bentuk normal pertama dicapai bila tiap nilai atribut adalah tunggal. Kondisi ini dapat diperoleh dengan melakukan eliminasi terjadinya data ganda. Pada kondisi normal pertama ini kemungkinan masih terjadi adanya data rangkap.

2. *Normalisasi kedua (2NF)*

Menurut Connolly dan Begg (2005,p407), normalisasi kedua merupakan sebuah relasi dalam 1NF dan setiap atribut *non-primary key* bersifat *full function dependency* pada *primary key* dari relasi tersebut. Dalam normalisasi kedua ini, atribut yang tergantung pada sebagian dari suatu *composite key* sebuah tabel dipindahkan ke sebuah tabel yang terpisah.

3. *Normalisasi ketiga (3NF)*

Menurut Connolly dan Begg (2005,p409) adalah sebuah relasi yang memenuhi normal pertama dan normal kedua dimana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* dari *primary key*-nya.

Suatu relasi dikatakan sudah mencapai bentuk normal ketiga jika :

- Berada dalam bentuk normal pertama dan kedua;
- Setiap atribut *non-primary key* tidak memiliki ketergantungan transitive kepada *primary key*.

2.1.19 Data flow Diagram (DFD)

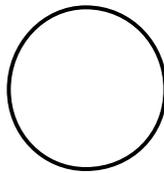
Data flow diagram merupakan salah satu *tools* yang dapat digunakan untuk menganalisis suatu *system*. DFD dapat digunakan

untuk mempresentasikan suatu sistem yang otomatis maupun manual, melalui gambar berbentuk jaringan grafik.

Simbol yang digunakan adalah sebagai berikut :



Eksternal *entity* (*entity* yang berada di luar sistem yang memberikan data kepada sistem atau yang menerima informasi dari sistem).



Proses (menggambarkan apa yang dilakukan oleh sistem)



Data flow (menggambarkan aliran data dari suatu *entity*



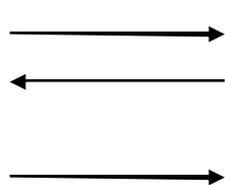
ke *entity* yang lain)



Data Store (tempat penyimpanan data, proses dapat mengambil data dari atau memberikan data ke *data store*).

2.1.20 *State Transition Diagram* (STD)

Merupakan *tools* pemodelan yang menggambarkan sifat ketergantungan pada waktu dari suatu sistem. Simbol-simbol yang digunakan antara lain



State/Keadaan

Perubahan *state* / keadaan

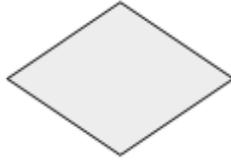
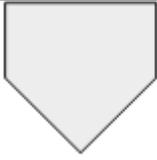
Untuk melengkapi STD masih diperlukan dua hal lagi, yaitu :

1. *Condition* adalah sebuah sinyal yang menyebabkan perubahan terhadap *state* dari *state* satu ke *state* berikutnya.
2. *Action* adalah sebuah aksi yang akan dilakukan sistem bila terjadi perubahan *state* (dimana merupakan reaksi terhadap kondisi).

2.1.21 Diagram Aliran Dokumen

Menurut Bodnar dan Hopwood (2000, pp37-48), Diagram Alir Dokumen merupakan suatu diagram simbolik yang menunjukkan aliran data dan urutan operasi dalam suatu sistem, seperti terlihat pada Tabel 2.1.

Tabel 2.1 Simbol Diagram Alir Dokumen

Simbol	Nama	Keterangan
	Terminal	Simbol ini digunakan untuk menunjukkan sumber atau tujuan data.
	Dokumen	Simbol ini digunakan untuk melihat setiap dokumen yang digunakan dalam sistem aplikasi dan mengidentifikasi titik mulai, distribusi dan disposisi.
	Proses manual	Simbol ini mewakili setiap proses <i>off-line</i> yang sesuai dengan kecepatan manusia, tanpa menggunakan bantuan mesin.
	Keputusan	Simbol ini digunakan untuk mewakili keputusan atau perubahan jenis operasi yang menentukan sejumlah alternatif jalur mana yang harus diikuti.
	Penghubung Lain Halaman	Simbol ini mewakili masuk dan keluar dari satu halaman ke halaman lain dari bagan aliran.
	Garis arus	Simbol ini mengindikasikan aliran dari data.

2.1.22 *Sistem Definition (Definisi Sistem)*

Definisi sistem mendeskripsikan jangkauan dan batasan dari aplikasi basis data.

Sebelum merancang aplikasi basis data, diperlukan identifikasi terhadap batasan dari sistem dan bagaimana antar mukanya dengan bagian lain dari sistem informasi organisasi. Sistem yang dibuat, diusahakan agar tidak hanya mencakup pengguna dan aplikasi yang sekarang, tetapi juga pengguna dan aplikasi untuk masa yang akan datang.

Aplikasi basis data dapat memiliki satu atau lebih *user views*. *User views* menentukan apa yang dibutuhkan oleh aplikasi basis data dari perspektif peranan pekerjaan tertentu dan area aplikasi perusahaan. Mengidentifikasi pandangan pengguna tersebut merupakan aspek penting dalam mengembangkan aplikasi basis data karena dapat membantu menyakinkan bahwa tidak ada pengguna utama yang terlupakan ketika sedang mengembangkan kebutuhan untuk aplikasi baru. *User views* juga membantu dalam pengembangan aplikasi basis data yang kompleks menurut (Connolly dan Begg, 2005, p286-287).

2.1.23 Requirement Collection and Analysis (Pengumpulan dan Analisis Kebutuhan)

Menurut Connolly dan Begg (2005, p288), pengumpulan dan analisis kebutuhan adalah proses pengumpulan analisis informasi tentang bagian perusahaan yang didukung oleh aplikasi basis data dan yang menggunakan informasi ini untuk mengidentifikasi kebutuhan-kebutuhan user dari sistem yang baru.

Pengumpulan kebutuhan dan *fact-finding* memiliki teknik yang penggunaanya sesuai kebutuhan :

1. Memeriksa dokumentasi

Pemahaman terhadap jalannya sistem akan cepat diperoleh dengan memeriksa dokumen-dokumen, laporan dan file yang berhubungan dengan sistem yang sedang berjalan.

2. Wawancara

Menurut Connolly and Begg (2005, p317), wawancara bertujuan untuk mengumpulkan fakta-fakta, memeriksa kebenaran fakta yang ada dan mengklarifikasinya untuk membangkitkan semangat, melibatkan pengguna akhir, mengidentifikasi kebutuhan-kebutuhan, dan mengumpulkan ide-ide dan pendapat.

3. Mengamati Operasional Perusahaan

Memungkinkan untuk ikut serta atau mengamati seseorang dalam melakukan kegiatan untuk mempelajari sistem. Salah satu faktor

pengamatan dapat berhasil adalah dengan mencari informasi sebanyak mungkin tentang aktivitas yang akan diamati serta orang yang melakukan aktivitas tersebut.

4. Penelitian

Jurnal komputer, buku-buku referensi, dan internet merupakan sumber informasi yang baik yang menyediakan informasi bagaimana orang lain memecahkan masalah yang serupa.

5. Kuisisioner

Melakukan survei melalui daftar pertanyaan. Daftar pertanyaan adalah special-purpose dokumen yang mengijinkan fakta untuk dikumpulkan dari sejumlah besar.

2.1.24 Delapan Aturan Emas Perancangan Interface

Menurut Shneiderman, Ben (2003, p89), ada delapan aturan emas dalam perancangan *interface*, antara lain :

1. Berusaha untuk konsisten;
2. Memungkinkan *frequent users* menggunakan *shortcuts*;
3. Memberikan umpan balik yang *informative*;
4. Merancang dialog yang memberikan penutupan (keadaan akhir);
5. Memberikan pencegahan kesalahan dan penanganan kesalahan yang sederhana;
6. Memungkinkan pembalikan aksi yang mudah;

7. Mendukung pusat kendali internal;
8. Mengurangi beban ingatan jangka pendek.

2.2 Teori-teori Khusus yang Berhubungan dengan Topik yang Dibahas

2.2.1 Pengertian Sewa-Menyewa

Menurut Anonimusc(2009) <http://organisasi.org>, sewa menyewa adalah suatu perjanjian atau kesepakatan di mana penyewa harus membayarkan atau memberikan imbalan atau manfaat dari benda atau barang yang dimiliki oleh pemilik barang yang dipinjamkan. Hukum dari sewa menyewa adalah mubah atau diperbolehkan. Contoh sewa menyewa dalam kehidupan sehari-hari misalnya seperti kontrak mengontrak gedung kantor, sewa lahan tanah untuk pertanian, menyewa / carter kendaraan, sewa menyewa vcd dan dvd original, dan lain-lain.

Sedangkan menurut Kamus Besar Bahasa Indonesia (Departemen Pendidikan dan Kebudayaan Republik Indonesia, 1988, p833), sewa menyewa adalah pemakaian sesuatu dengan membayar uang sewa, uang yang dibayarkan karena memakasi atau meminjam sesuatu, yang boleh dipakai dengan membayar dengan uang. Sedangkan pengertian penyewaan adalah proses, cara, perbuatan menyewa atau menyewakan.

- a. Hal-hal yang Membuat Sewa Menyewa Batal
 - Barang yang disewakan rusak;

- Periode / masa perjanjian / kontrak sewa menyewa telah habis;
 - Barang yang disewakan cacat setelah berada di tangan penyewa.
- b. Manfaat Sewa Menyewa
- Membantu orang lain yang tidak sanggup membeli barang;
 - Yang menyewakan mendapatkan manfaat dari sang penyewa.

2.2.2 Perjanjian Kerjasama

A. Pengertian Perjanjian

Menurut Anonimud(2009) <http://www.lexregis.com>,

Suatu perjanjian adalah suatu peristiwa dimana seseorang berjanji kepada seorang lain atau dimana dua orang itu saling berjanji untuk melaksanakan sesuatu hal.

B. Sahnya Perjanjian

Suatu perjanjian dinyatakan sah apabila memenuhi beberapa syarat, yaitu:

1. Berdasarkan kesepakatan para pihak
2. Pihak-pihak dalam perjanjian harus cakap untuk membuat perjanjian

Berikut adalah pihak-pihak yang tidak cakap secara hukum untuk membuat perjanjian:

- a. Orang yang belum dewasa, yaitu orang yang belum berumur 21 tahun;
 - b. Orang-orang yang ditaruh dibawah pengampuan, misalnya: anak-anak, orang yang pikirannya kurang sehat atau mengalami gangguan mental;
 - c. Semua pihak yang menurut undang-undang yang berlaku tidak cakap atau dibatasi kecakapannya untuk membuat perjanjian, misalnya; istri dalam melakukan perjanjian untuk transaksi-transaksi tertentu harus mendapatkan persetujuan suami;
3. Perjanjian menyepakati suatu hal;
 4. Dibuat berdasarkan suatu sebab yang halal.

C. Sebab-sebab Berakhirnya Perjanjian

Terpenuhinya prestasi atau perikatan yang disepakati dan syarat-syarat tertentu dalam perjanjian dapat menjadi sebab berakhirnya perjanjian, misalnya habisnya jangka waktu yang telah disepakati dalam perjanjian atau dalam *loan agreement*, semua hutang dan bunga atau denda jika ada telah dibayarkan. Secara keseluruhan, KUHPerdato mengatur faktor-faktor lain

yang dapat menyebabkan berakhirnya perjanjian, diantaranya karena:

1. Pembayaran;
2. Penawaran pembayaran, diikuti dengan penyimpanan atau penitipan;
3. Pembaharuan hutang;
4. Perjumpaan Hutang atau kompensasi;
5. Percampuran Hutang;
6. Pembebasan Hutang;
7. Musnahnya barang yang terhutang;
8. Kebatalan atau pembatalan;
9. Berlakunya suatu syarat batal;
10. Lewatnya waktu.

2.3 Teori Pendukung

2.3.1 Pengenalan PHP

PHP merupakan singkatan dari *PHP Hypertext Preprocessor* yaitu bahasa berbentuk skrip yang disimpan dan dijalankan pada sisi server. Sedangkan hasil (output) eksekusi dari server ditampilkan pada client dengan menggunakan web browser.

Penggunaan PHP biasanya difokuskan pada pengembangan aplikasi yang server-side scripting. Namun sebenarnya terdapat beberapa area utama penggunaan PHP, di antaranya :

- *Server-side scripting*;
- *Commandline scripting*;
- *Desktop application* (menggunakan PHPGTK).

Dari semua penggunaan PHP, penggunaan pada *server-side scripting* merupakan yang paling sering digunakan. Terutama bila membutuhkan website atau aplikasi berbasis web yang dinamis

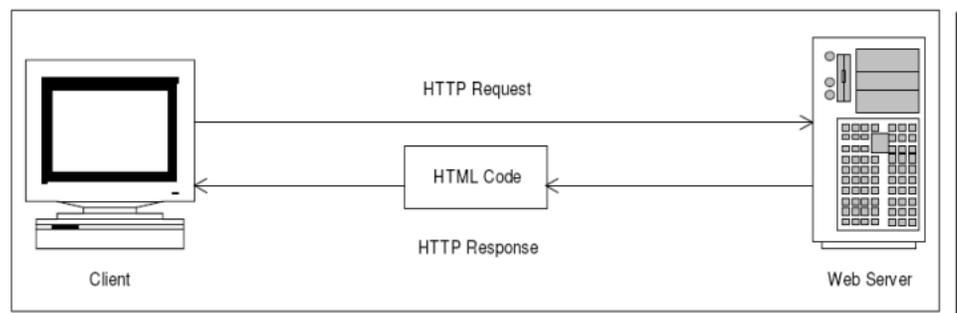
PHP dapat dijalankan pada hampir semua sistem operasi yang ada saat ini, di antaranya : Linux, Unix (HPUX, Solaris, OpenBSD), Microsoft Windows, Mac OS X, RISC OS, dan lainlain.

PHP juga mendukung banyak web server, di antaranya : Apache, Microsoft Internet Information Server (IIS), Personal Web Server (PWS), Netscape server, iPlanet server, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPD, dan lain-lain.

Sedangkan dalam melakukan penulisan program PHP dapat menggunakan *procedural programming* atau *object oriented programming*. Selain itu, dapat juga menggunakan gabungan keduanya. Dengan menggunakan PHP output (hasil keluaran) tidak harus berupa HTML, namun PHP mempunyai kemampuan untuk menghasilkan gambar, PDF bahkan file Flash yang dihasilkan secara *on fly*.

PHP adalah sebuah script pemrograman yang terletak dan dieksekusi di server. Salah satu fungsinya adalah untuk menerima, mengolah, dan menampilkan data dari dan ke sebuah situs. Data yang diterima akan diolah di sebuah program database server (program database yang terletak di sisi server, MySQL contohnya) untuk kemudian hasilnya ditampilkan kembali ke layar browser sebuah situs.

2.3.2 Konsep Kerja PHP



Gambar 2.4 : Konsep kerja HTML

Secara umum konsep kerja dari PHP hampir sama dengan konsep kerja dari HTML. Dimana terdapat *client* yang meminta (request) berkas tertentu yang disimpan pada sisi server. Kemudian server tersebut mengirimkan berkas tersebut kepada *client*. Perbedaannya adalah pada konsep kerja HTML berkas yang dikirimkan sama dengan yang disimpan pada sisi server. Sedangkan pada konsep kerja PHP, berkas yang dikirim merupakan hasil proses pada server sehingga bisa terdapat perbedaan

antara *source code* yang disimpan pada sisi server dengan yang dikirim pada *client*.

2.3.3 My SQL

Menurut Anonimuse(2009) <http://kajianwebsite.org>, MySQL adalah salah satu jenis program database server, yaitu sebuah program yang berfungsi untuk mengolah, menyimpan dan memanipulasi data di server.

Menurut Peranganin (2006, pp389-390), MySQL menggunakan suatu format standar SQL bahasa data yang terkenal. MySQL bekerja pada berbagai sistem operasi dan banyak bahasa. PHP menyediakan banyak fungsi untuk mendukung basis data MySQL.

2.3.4 Microsoft Visio

Menurut Anonimusf (2009) <http://miftahsapro.blogspot.com>, Microsoft Visio adalah salah satu program yang dapat digunakan untuk membuat diagram. Visio menyediakan banyak fasilitas yang membantu dalam pembuatan diagram untuk menggambarkan informasi dan sistem dari penjelasan dalam bentuk teks menjadi suatu diagram dalam bentuk gambar disertai penjelasan singkat.

Oleh karena Visio merupakan salah satu produk dari Microsoft Office, maka Visio dapat berkolaborasi dengan produk Microsoft Office

lainnya seperti Microsoft Word, Excel, PowerPoint, dan Project. Bentuk kolaborasi dapat dilakukan dengan fasilitas export dan import data.

2.3.5 PHP My Admin

Menurut Dodit Suprianto (2008, p225) Merupakan salah satu kakas pengola database MySQL yang berbasis web. PHP MyAdmin memberikan kemudahan dalam pengoperasiaanya. Hampir semua web *hosting* menyediakan php my admin untuk para penyewa *virtual host*.

2.3.6 Adobe Flash CS3

Adobe flash CS3 professional atau biasa disebut flash CS3 adalah program animasi populer berbasis vector yang digunakan untuk membuat animasi dan aplikasi web interaktif. Selain itu flash CS3 juga dapat digunakan untuk membuat game, website, menu interaktif, e-card, aplikasi multimedia hingga aplikasi untuk ponsel

2.3.7 Word Wide Web (WWW)

Menurut Turban et.al. (2007, p142), *World Wide Web (WWW)* atau yang dikenal sebagai W3 adalah sebuah sistem yang telah disepakati secara universal sebagai standar untuk menampung, mengambil, memformat, dan menampilkan informasi melalui sebuah *client/server architecture*.

2.3.8 *Computer Supported Cooperative Work (CSCW)*

Menurut Indra Yatini (2007, p97) *Computer Supported Cooperative Work (CSCW)* adalah bidangn studi yang berfokus pada perancangan dan evaluasi teknologi baru untuk mendukung proses sosial kerja, sering diantara mitra yang berjauhan.