

## BAB 2

### LANDASAN TEORI

#### 2.1. Pendekatan Basisdata

##### 2.1.1. Pengertian Basisdata

Menurut Indrajani (2009, p2) beberapa pengertian basis data, yaitu :

- a) Kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengkonsolidasi banyak catatan yang sebelumnya disimpan dalam *file* terpisah.
- b) Merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi. Artinya, basis data merupakan tempat penyimpanan data besar yang dapat digunakan oleh banyak pengguna.

Menurut Connolly & Begg (2005,p15), basis data adalah suatu kumpulan logika data yang terhubung satu sama lain, dan deskripsi dari data ini dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

Menurut Date (2000,p10), basis data merupakan kumpulan dari data yang hampir tidak mengalami perubahan dan digunakan oleh aplikasi sistem pada beberapa perusahaan.

### 2.1.2. Database Management System

Menurut Connolly & Begg (2005,p16) *Database Management System* (DBMS) merupakan sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, mengatur, dan mengontrol pengaksesan ke basis data.

#### 2.1.2.1 Komponen DBMS

Menurut Connolly (2005,p19) ada lima komponen utama di dalam ruang lingkup DBMS, yaitu :

1. *Hardware* (Perangkat Keras)

DBMS dan aplikasi membutuhkan perangkat keras agar dapat dijalankan. Perangkat keras dapat mencakup *single personal computer, single mainframe*, atau sebuah jaringan komputer.

2. *Software* (Perangkat Lunak)

Komponen perangkat lunak terdiri dari perangkat lunak DBMS itu sendiri dan program aplikasi, bersama dengan sistem operasi, termasuk perangkat lunak jaringan jika DBMS digunakan dalam sebuah jaringan.

### 3. Data

Merupakan komponen yang paling penting di dalam ruang lingkup DBMS, berperan sebagai penghubung antara komponen mesin dan manusia.

### 4. Prosedur

Prosedur mencakup instruksi-instruksi dan aturan-aturan yang harus disertakan dalam mendesain dan menggunakan database dan DBMS.

### 5. Manusia

Komponen terakhir adalah manusia yang terlibat dengan sistem. Ada lima tipe berbeda yang terlibat dalam ruang lingkup DBMS, yaitu :

#### a. *Data Administrator*

*Data Administrator* (DA) bertanggung jawab untuk mengatur sumber daya meliputi perencanaan basis data, standar pengaturan dan pengembangan, membuat keputusan strategis dan kebijakan mengenai data yang ada.

#### b. *Database Administrator*

*Database Administrator* (DBA) bertanggung jawab dalam realisasi fisik basis data, meliputi rancangan fisik basis data dan implementasi, keamanan dan pengaturan integritas, menjaga sistem operasional, dan memastikan kinerja aplikasi untuk kepuasan pengguna.

c. *Database Designers*

*Logical database designer* berhubungan dengan identifikasi data (entity dan atribut), *relationship* antara data, dan *constraint* pada data untuk disimpan di basis data. *Physical database designer* memutuskan bagaimana rancangan basis data logikal dapat direalisasikan.

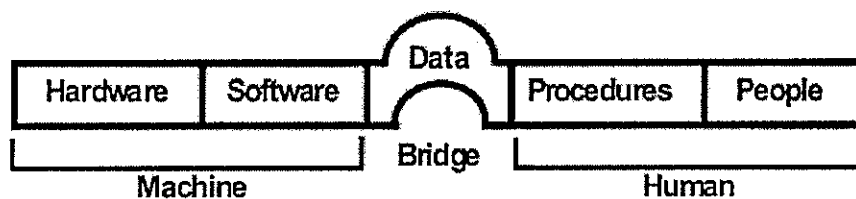
d. *Application programmers*

Bertanggung jawab untuk mengimplementasikan program aplikasi yang menyediakan syarat-syarat fungsionalitas untuk pengguna terakhir.

e. *End Users*

*End users* adalah *client* bagi basis data, yang telah dirancang dan diimplementasikan dan dijaga untuk menyediakan kebutuhan informasi mereka.

Komponen-komponen DBMS dapat digambarkan sebagai berikut :



Gambar 2.1 Komponen DBMS (Connolly, 2005, p19)

### 2.1.3. *Structure Query Language*

#### 2.1.3.1 *Data Definition Language*

Menurut Connolly(2005, p40) *Data Definition Language* (DDL) adalah sebuah bahasa yang memungkinkan DBA atau *user* untuk menggambarkan dan menentukan nama dari entitas, atribut, dan hubungan yang diperlukan untuk sebuah aplikasi, bersamaan dengan integritas yang terasosiasi dan batasan keamanan yang ada.

Jadi, dapat disimpulkan bahwa *Data Definition Language* (DDL) adalah sebuah bahasa yang memungkinkan *user* untuk melihat database, maupun menggambarkan serta menentukan nama dari entitas, atribut, dan hubungan yang diperlukan untuk sebuah aplikasi.

#### 2.1.3.2 *Data Manipulation Language*

Menurut Connolly (2005, p40) *Data Manipulation Language* (DML) adalah sebuah bahasa yang memberikan fasilitas pengoperasian terhadap data yang ada di dalam basis data.

DML biasanya meliputi:

1. Memasukkan data yang baru ke dalam basis data.
2. Memodifikasi data yang tersimpan di dalam basis data.
3. Mengambil data yang terdapat di dalam basis data.
4. Menghapus data dari basis data.

- b. *Report Generators* merupakan fasilitas yang berfungsi untuk menghasilkan laporan dari data yang disimpan dalam basis data.
- c. *Graphic Generators* merupakan fasilitas yang berfungsi mengambil data dari basis data dan menampilkannya dalam bentuk grafik yang
- d. menunjukkan hubungan dalam data.
- e. *Application Generators* merupakan fasilitas yang menghasilkan program yang berhubungan dengan basis data.

#### **2.1.5. Database System Development Lifecycle (DSDLC)**

Menurut Connolly dan Begg (2005, p283), sistem basis data adalah komponen dasar dari organisasi yang besar dengan sistem informasi yang luas. Hal penting yang perlu diperhatikan dalam *Database Application Lifecycle* adalah bahwa tingkatannya tidak sepenuhnya berurutan (*sequential*). Dimana ada beberapa tingkatan yang berulang dengan alur-balik (*feedback loop*) misalnya, masalah ditemukan pada tingkatan perancangan basis data yang membutuhkan tambahan kumpulan kebutuhan dan analisis.

#### **2.1.5.1. Database Planning**

Merupakan aktivitas manajemen yang memungkinkan tahapan dari *database application lifecycle* direalisasikan dan seefisien mungkin. Perencanaan *database* harus terintegrasi dengan keseluruhan strategi sistem informasi dari organisasi.

#### **2.1.5.2. System Definition**

Menjelaskan batasan-batasan dan cakupan dari aplikasi *database* dan sudut pandang *user* (*user view*) yang utama. *User view* mendefinisikan apa yang diwajibkan dari suatu aplikasi *database* dari perspektif aturan kerja khusus (seperti Manajer atau pengawas) atau area aplikasi perusahaan (seperti *Marketing*, *Personnal*, atau *Stock Control*).

Aplikasi *database* dapat memiliki satu atau lebih *user view*. Identifikasi *user view* membantu memastikan bahwa tidak ada *user* utama dari suatu *database* yang terlupakan ketika pembuatan aplikasi baru. *User view* juga membantu dalam pengembangan aplikasi *database* yang rumit/kompleks dengan memecah permintaan-permintaan ke dalam bagian-bagian yang lebih sederhana.

#### **2.1.5.3. Requirement Collection and Analysis**

Suatu proses pengumpulan dan analisa informasi mengenai bagian organisasi yang akan didukung oleh aplikasi *database*, dan menggunakan informasi tersebut untuk identifikasi kebutuhan *user* akan sistem baru. Informasi dianalisa untuk mengidentifikasi kebutuhan agar disertakan dalam aplikasi

### c. *Physical Database Design*

Suatu proses yang menghasilkan dekripsi implementasi *database* pada penyimpanan sekunder. Menggambarkan struktur penyimpanan dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Dapat dikatakan juga, desain model data fisikal merupakan cara pembuatan menuju sistem DBMS tertentu.

#### 2.1.5.5. *DBMS Selection*

Yaitu fase pemilihan DBMS yang tepat untuk mendukung aplikasi *database*. Dapat dilakukan kapanpun sebelum menuju desain *logical* asalkan terdapat cukup informasi mengenai kebutuhan sistem.

Tahapan dalam memilih DBMS yang tepat antara lain :

1. Mendefinisikan syarat-syarat sebagai referensi.
2. Daftar singkat dari dua atau tiga produk.
3. Mengevaluasi produk-produk.
4. Merekomendasikan pilihan dan membuat laporan.

#### 2.1.5.6. *Application Design*

Yaitu proses membuat/mendesain *user interface* dan program aplikasi yang menggunakan dan memproses *database*. Desain *database* dan aplikasi merupakan aktivitas parallel yang meliputi dua aktivitas penting yaitu:



## BAB 2

### LANDASAN TEORI

#### 2.1. Pendekatan Basisdata

##### 2.1.1. Pengertian Basisdata

Menurut Indrajani (2009, p2) beberapa pengertian basis data, yaitu :

- a) Kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengkonsolidasi banyak catatan yang sebelumnya disimpan dalam *file* terpisah.
- b) Merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi. Artinya, basis data merupakan tempat penyimpanan data besar yang dapat digunakan oleh banyak pengguna.

Menurut Connolly & Begg (2005,p15), basis data adalah suatu kumpulan logika data yang terhubung satu sama lain, dan deskripsi dari data ini dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

Menurut Date (2000,p10), basis data merupakan kumpulan dari data yang hampir tidak mengalami perubahan dan digunakan oleh aplikasi sistem pada beberapa perusahaan.

### 2.1.2. Database Management System

Menurut Connolly & Begg (2005,p16) *Database Management System* (DBMS) merupakan sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, mengatur, dan mengontrol pengaksesan ke basis data.

#### 2.1.2.1 Komponen DBMS

Menurut Connolly (2005,p19) ada lima komponen utama di dalam ruang lingkup DBMS, yaitu :

1. *Hardware* (Perangkat Keras)

DBMS dan aplikasi membutuhkan perangkat keras agar dapat dijalankan. Perangkat keras dapat mencakup *single personal computer, single mainframe*, atau sebuah jaringan komputer.

2. *Software* (Perangkat Lunak)

Komponen perangkat lunak terdiri dari perangkat lunak DBMS itu sendiri dan program aplikasi, bersama dengan sistem operasi, termasuk perangkat lunak jaringan jika DBMS digunakan dalam sebuah jaringan.

### 3. Data

Merupakan komponen yang paling penting di dalam ruang lingkup DBMS, berperan sebagai penghubung antara komponen mesin dan manusia.

### 4. Prosedur

Prosedur mencakup instruksi-instruksi dan aturan-aturan yang harus disertakan dalam mendesain dan menggunakan database dan DBMS.

### 5. Manusia

Komponen terakhir adalah manusia yang terlibat dengan sistem. Ada lima tipe berbeda yang terlibat dalam ruang lingkup DBMS, yaitu :

#### a. *Data Administrator*

*Data Administrator* (DA) bertanggung jawab untuk mengatur sumber daya meliputi perencanaan basis data, standar pengaturan dan pengembangan, membuat keputusan strategis dan kebijakan mengenai data yang ada.

#### b. *Database Administrator*

*Database Administrator* (DBA) bertanggung jawab dalam realisasi fisik basis data, meliputi rancangan fisik basis data dan implementasi, keamanan dan pengaturan integritas, menjaga sistem operasional, dan memastikan kinerja aplikasi untuk kepuasan pengguna.

c. *Database Designers*

*Logical database designer* berhubungan dengan identifikasi data (entity dan atribut), *relationship* antara data, dan *constraint* pada data untuk disimpan di basis data. *Physical database designer* memutuskan bagaimana rancangan basis data logikal dapat direalisasikan.

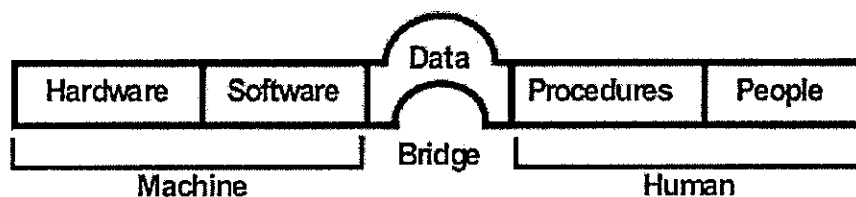
d. *Application programmers*

Bertanggung jawab untuk mengimplementasikan program aplikasi yang menyediakan syarat-syarat fungsionalitas untuk pengguna terakhir.

e. *End Users*

*End users* adalah *client* bagi basis data, yang telah dirancang dan diimplementasikan dan dijaga untuk menyediakan kebutuhan informasi mereka.

Komponen-komponen DBMS dapat digambarkan sebagai berikut :



**Gambar 2.1** Komponen DBMS (Connolly, 2005, p19)

### 2.1.3. *Structure Query Language*

#### 2.1.3.1 *Data Definition Language*

Menurut Connolly(2005, p40) *Data Definition Language* (DDL) adalah sebuah bahasa yang memungkinkan DBA atau *user* untuk menggambarkan dan menentukan nama dari entitas, atribut, dan hubungan yang diperlukan untuk sebuah aplikasi, bersamaan dengan integritas yang terasosiasi dan batasan keamanan yang ada.

Jadi, dapat disimpulkan bahwa *Data Definition Language* (DDL) adalah sebuah bahasa yang memungkinkan *user* untuk melihat database, maupun menggambarkan serta menentukan nama dari entitas, atribut, dan hubungan yang diperlukan untuk sebuah aplikasi.

#### 2.1.3.2 *Data Manipulation Language*

Menurut Connolly (2005, p40) *Data Manipulation Language* (DML) adalah sebuah bahasa yang memberikan fasilitas pengoperasian terhadap data yang ada di dalam basis data.

DML biasanya meliputi:

1. Memasukkan data yang baru ke dalam basis data.
2. Memodifikasi data yang tersimpan di dalam basis data.
3. Mengambil data yang terdapat di dalam basis data.
4. Menghapus data dari basis data.

### ***Data Control Language***

*Data Control Languages* (DCL) memperbolehkan pengguna untuk menjaga agar hanya *user* yang memiliki hak akses yang dapat melakukan suatu aksi.

#### **2.1.4. *Fourth Generation Language***

4GL (*Fourth-Generation Languages*) adalah sebuah bahasa non-prosedural dimana pengguna mendefinisikan apa yang harus dilakukan bukan bagaimana sesuatu dilakukan.

*Fourth-Generation Languages* meliputi :

- a. *Presentation Languages*, seperti *query languages* dan *report generators*.
- b. *Speciality Languages*, seperti *spreadsheets* dan *database languages*.
- c. *Application Generators*, yang terdiri dari *insert*, *update*, *select* untuk membuat aplikasi.
- d. *Very High Level Languages*, yang digunakan untuk menghasilkan *application code*.

Tipe-tipe dari 4GL :

- a. *Forms Generators* merupakan fasilitas interaktif untuk melakukan inputan data dan *display layouts* pada layar.

- b. *Report Generators* merupakan fasilitas yang berfungsi untuk menghasilkan laporan dari data yang disimpan dalam basis data.
- c. *Graphic Generators* merupakan fasilitas yang berfungsi mengambil data dari basis data dan menampilkannya dalam bentuk grafik yang
- d. menunjukkan hubungan dalam data.
- e. *Application Generators* merupakan fasilitas yang menghasilkan program yang berhubungan dengan basis data.

#### **2.1.5. Database System Development Lifecycle (DSDLC)**

Menurut Connolly dan Begg (2005, p283), sistem basis data adalah komponen dasar dari organisasi yang besar dengan sistem informasi yang luas. Hal penting yang perlu diperhatikan dalam *Database Application Lifecycle* adalah bahwa tingkatannya tidak sepenuhnya berurutan (*sequential*). Dimana ada beberapa tingkatan yang berulang dengan alur-balik (*feedback loop*) misalnya, masalah ditemukan pada tingkatan perancangan basis data yang membutuhkan tambahan kumpulan kebutuhan dan analisis.

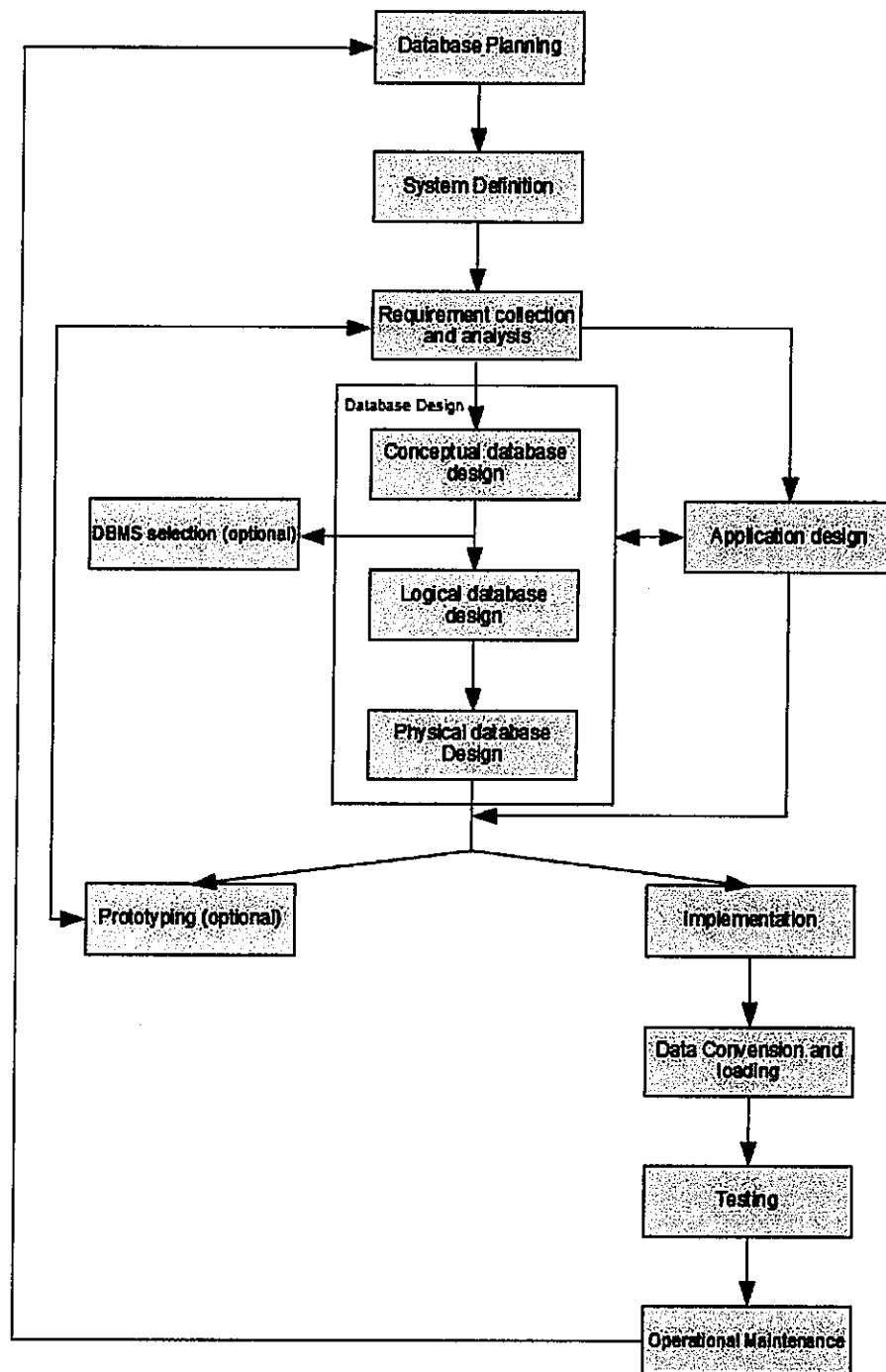
- b. *Report Generators* merupakan fasilitas yang berfungsi untuk menghasilkan laporan dari data yang disimpan dalam basis data.
- c. *Graphic Generators* merupakan fasilitas yang berfungsi mengambil data dari basis data dan menampilkannya dalam bentuk grafik yang
- d. menunjukkan hubungan dalam data.
- e. *Application Generators* merupakan fasilitas yang menghasilkan program yang berhubungan dengan basis data.

#### **2.1.5. Database System Development Lifecycle (DSDLC)**

Menurut Connolly dan Begg (2005, p283), sistem basis data adalah komponen dasar dari organisasi yang besar dengan sistem informasi yang luas. Hal penting yang perlu diperhatikan dalam *Database Application Lifecycle* adalah bahwa tingkatannya tidak sepenuhnya berurutan (*sequential*). Dimana ada beberapa tingkatan yang berulang dengan alur-balik (*feedback loop*) misalnya, masalah ditemukan pada tingkatan perancangan basis data yang membutuhkan tambahan kumpulan kebutuhan dan analisis.



Berikut ini adalah tahapan dari *database lifecycle* menurut Connolly (2005,p283-306):



**Gambar 2.2 Tahapan Database Lifecycle(Connolly, 2005, p284)**

#### **2.1.5.1. Database Planning**

Merupakan aktivitas manajemen yang memungkinkan tahapan dari *database application lifecycle* direalisasikan dan seefisien mungkin Perencanaan *database* harus terintegrasi dengan keseluruhan strategi sistem informasi dari organisasi.

#### **2.1.5.2. System Definition**

Menjelaskan batasan-batasan dan cakupan dari aplikasi *database* dan sudut pandang *user* (*user view*) yang utama. *User view* mendefinisikan apa yang diwajibkan dari suatu aplikasi *database* dari perspektif aturan kerja khusus (seperti Manajer atau pengawas) atau area aplikasi perusahaan (seperti *Marketing*, *Personnal*, atau *Stock Control*).

Aplikasi *database* dapat memiliki satu atau lebih *user view*. Indentifikasi *user view* membantu memastikan bahwa tidak ada *user* utama dari suatu *database* yang terlupakan ketika pembuatan aplikasi baru. *User view* juga membantu dalam pengembangan aplikasi *database* yang rumit/kompleks dengan memecah permintaan-permintaan ke dalam bagian-bagian yang lebih sederhana.

#### **2.1.5.3. Requirement Collection and Analysis**

Suatu proses pengumpulan dan analisa informasi mengenai bagian organisasi yang akan didukung oleh aplikasi *database*, dan menggunakan informasi tersebut untuk identifikasi kebutuhan *user* akan sistem baru. Informasi dianalisa untuk mengidentifikasi kebutuhan agar disertakan dalam aplikasi

*database* yang akan dibuat. Informasi dikumpulkan untuk setiap *user view* utama meliputi:

- a. Deskripsi data yang digunakan atau dihasilkan.
- b. Detail mengenai bagaimana data digunakan atau dihasilkan.
- c. Beberapa kebutuhan tambahan untuk aplikasi *database* yang baru.

#### 2.1.5.4. *Database Design*

Merupakan suatu proses pembuatan sebuah desain *database* yang akan mendukung tujuan dan operasi suatu organisasi/perusahaan.

Ada tiga fase dalam mendesain *database*, yaitu:

##### a. *Conceptual Database Design*

Suatu proses pembentukan model dari informasi yang digunakan dalam perusahaan/organisasi, independen dari keseluruhan aspek fisik. Model data dibangun dengan menggunakan informasi dalam spesifikasi kebutuhan *user*. Model data konseptual merupakan sumber informasi untuk fase *logical database design*.

##### b. *Logical Database Design*

Suatu proses pembentukan model dari informasi yang digunakan dalam perusahaan/organisasi berdasarkan model data tertentu (misal: relasional), tetapi independen terhadap DBMS tertentu dan aspek fisik lainnya. Model data konseptual yang telah dibuat sebelumnya, diperbaiki dan dipetakan kedalam model data *logical*.

### c. *Physical Database Design*

Suatu proses yang menghasilkan dekripsi implementasi *database* pada penyimpanan sekunder. Menggambarkan struktur penyimpanan dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Dapat dikatakan juga, desain model data fisik merupakan cara pembuatan menuju sistem DBMS tertentu.

#### 2.1.5.5. *DBMS Selection*

Yaitu fase pemilihan DBMS yang tepat untuk mendukung aplikasi *database*. Dapat dilakukan kapanpun sebelum menuju desain *logical* asalkan terdapat cukup informasi mengenai kebutuhan sistem.

Tahapan dalam memilih DBMS yang tepat antara lain :

1. Mendefinisikan syarat-syarat sebagai referensi.
2. Daftar singkat dari dua atau tiga produk.
3. Mengevaluasi produk-produk.
4. Merekomendasikan pilihan dan membuat laporan.

#### 2.1.5.6. *Application Design*

Yaitu proses membuat/mendesain *user interface* dan program aplikasi yang menggunakan dan memproses *database*. Desain *database* dan aplikasi merupakan aktivitas parallel yang meliputi dua aktivitas penting yaitu:

- 4) *Visually appealing layout of the form/report*, tampilan *form/report* harus menarik dan sesuai dengan *hardcopy* agar konsisten.
- 5) *Familiar field labels*, penggunaan label yang familiar.
- 6) *Consistency terminology and abbreviation*, terminologi dan singkatan yang digunakan harus konsisten.
- 7) *Consistency use of color*, penggunaan warna yang konsisten.
- 8) *Visible space and boundaries for data-entry fields*, jumlah tempat yang disediakan untuk data *entry* harus diketahui oleh *user*.
- 9) *Convenient cursor movement*, *user* dapat dengan mudah menjalankan operasi yang diinginkan dengan menggerakkan *cursor* pada *form/report*.
- 10) *Error correction for individual characters and entire fields*, *user* dengan mudah menjalankan operasi yang diinginkan dan melakukan perubahan terhadap nilai *field*.
- 11) *Error messages for unacceptable values*, menampilkan pesan *error* bila ada *input* yang tidak sesuai atau tidak dapat diterima.
- 12) *Optional fields marked clearly*, pemberian tanda pada *fields* yang *optional*.
- 13) *Explanatory messages for fields*, ketika *user* meletakkan *cursor* pada suatu *fields*, maka keterangan mengenai *field* tersebut harus dapat terlihat.
- 14) *Completion signal*, indikator yang menjelaskan bahwa suatu proses telah selesai dilaksanakan.

#### 2.1.5.7. *Prototyping*

*Prototyping* adalah membuat model kerja dari suatu aplikasi *database*.

Tujuan utama dari *prototyping* adalah:

- a. Untuk mengidentifikasi *feature* dari sistem yang berjalan dengan baik atau tidak.
- b. Untuk memberikan perbaikan-perbaikan atau penambahan *feature* baru.
- c. Untuk klarifikasi kebutuhan *user*.
- d. Untuk evaluasi feasibilitas (kemungkinan yang akan terjadi) dari desain sistem khusus.

#### 2.1.5.8. *Implementation*

Merupakan realisasi fisik dari *database* dan desain aplikasi. Dalam tahap ini juga akan diimplementasikan komponen lain dari aplikasi basis data seperti menu layar, pemasukan data, keamanan dan control integritas.

#### 2.1.5.9. *Data Conversion and Loading*

Pemindahan data yang ada ke dalam *database* baru dan mengkonversikan aplikasi yang ada agar dapat digunakan pada *database* yang baru. Tahapan ini dibutuhkan ketika sistem *database* baru menggantikan sistem yang lama. DBMS biasanya memiliki utilitas yang memanggil ulang *file* yang sudah ada ke dalam *database* baru.

#### 2.1.5.10. *Testing*

Suatu proses eksekusi program aplikasi dengan tujuan untuk menemukan kesalahan. Dengan menggunakan strategi tes yang direncanakan dan data yang sesungguhnya. Pengujian tidak dapat menunjukkan kesalahan, pengujian hanya akan terlihat jika terjadi kesalahan perangkat lunak. Mendemonstrasikan *database* dan program aplikasi agar berjalan seperti yang diharapkan.

#### 2.1.5.11. *Operational Maintenance*

Suatu proses pengawasan dan pemeliharaan sistem setelah instalasi, meliputi:

- a. Pengawasan performa sistem, jika performa menurun maka memerlukan perbaikan atau pengaturan ulang *database*.
- b. Pemeliharaan dan pembaharuan aplikasi *database* (jika dibutuhkan).
- c. Penggabungan kebutuhan baru ke dalam aplikasi *database*.

### 2.1.6 *Entity - Relationship Modeling*

Model *Entity-Relationship* merupakan salah satu model yang dapat memastikan pemahaman yang tepat terhadap data dan bagaimana penggunaannya di dalam suatu organisasi (Connolly, 2005, p342). Model ini dimulai dengan identifikasi entitas dan *relationship* antar data yang harus direpresentasikan di dalam model, dan kemudian ditambahkan atribut dan setiap *constraint* pada entitas, *relationship*, dan atributnya.

Simbol – simbol ERD :

#### 2.1.5.10. *Testing*

Suatu proses eksekusi program aplikasi dengan tujuan untuk menemukan kesalahan. Dengan menggunakan strategi tes yang direncanakan dan data yang sesungguhnya. Pengujian tidak dapat menunjukkan kesalahan, pengujian hanya akan terlihat jika terjadi kesalahan perangkat lunak. Mendemonstrasikan *database* dan program aplikasi agar berjalan seperti yang diharapkan.

#### 2.1.5.11. *Operational Maintenance*

Suatu proses pengawasan dan pemeliharaan sistem setelah instalasi, meliputi:

- a. Pengawasan performa sistem, jika performa menurun maka memerlukan perbaikan atau pengaturan ulang *database*.
- b. Pemeliharaan dan pembaharuan aplikasi *database* (jika dibutuhkan).
- c. Penggabungan kebutuhan baru ke dalam aplikasi *database*.

### 2.1.6 *Entity - Relationship Modeling*

Model *Entity-Relationship* merupakan salah satu model yang dapat memastikan pemahaman yang tepat terhadap data dan bagaimana penggunaannya di dalam suatu organisasi (Connolly, 2005, p342). Model ini dimulai dengan identifikasi entitas dan *relationship* antar data yang harus direpresentasikan di dalam model, dan kemudian ditambahkan atribut dan setiap *constraint* pada entitas, *relationship*, dan atributnya.

Simbol – simbol ERD :



### 1. Entitas

Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.

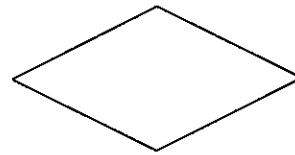
Simbol yang digunakan adalah :



### 2. Relasi

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.

Simbol yang digunakan adalah :



### 3. Garis

Garis sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

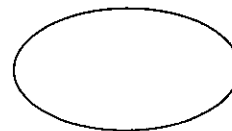
Simbol yang digunakan adalah :



### 4. Atribut

Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai *key* diberi garis bawah).

Simbol yang digunakan adalah :



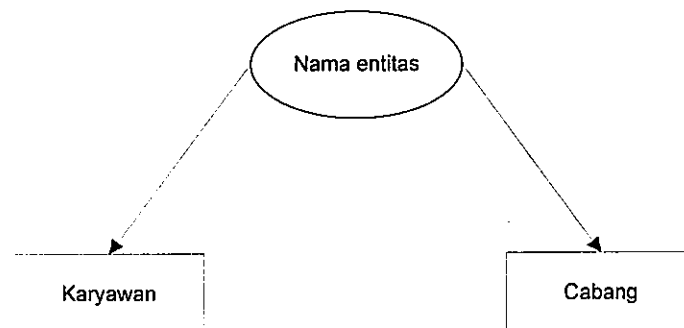
### 2.1.6.1 Entity Types

Menurut Connolly (2005, p343), entity adalah sekelompok objek dengan sifat yang sama, yang diidentifikasi oleh perusahaan sebagai memiliki keberadaan yang independen

Tipe entitas adalah sekumpulan objek yang memiliki properti yang sama, yang diidentifikasikan ke dalam organisasi karena keberadaannya yang bebas (*independent existence*) (Connolly, 2005, p343). Sedangkan *entity occurrence* adalah sebuah objek dari suatu tipe entitas yang dapat diidentifikasikan secara unik (Connolly, 2005, p345).

Keberadaan objek-objeknya secara fisik atau nyata (*physical existence*) seperti entitas pegawai, rumah, dan pelanggan, atau secara konseptual atau abstrak (*conceptual existence*) seperti entitas penjualan, pembelian, dan peminjaman.

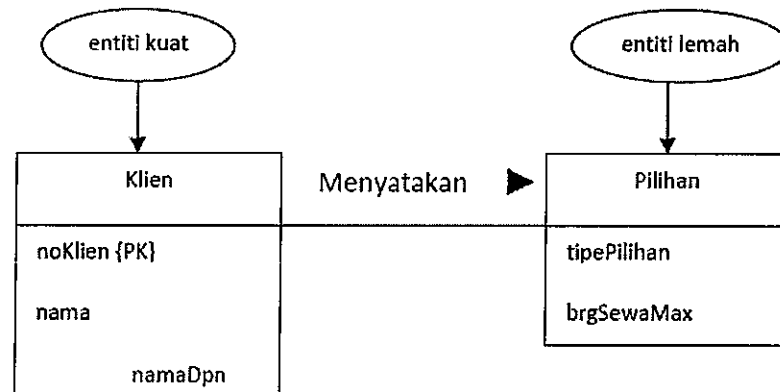
Setiap tipe entitas dilambangkan dengan sebuah persegi panjang yang diberi nama dari entitas tersebut. Nama tipe entitas biasanya adalah kata benda tunggal. Huruf pertama dari setiap kata pada nama tipe entitas ditulis dengan huruf besar.



Gambar 2.3 Contoh Tipe Entitas (Connolly and Begg, 2005, p345)

Tipe entitas dapat diklasifikasikan menjadi:

- a. Tipe Entitas Kuat, yaitu tipe entitas yang keberadaannya tidak tergantung pada tipe entitas lainnya (Connolly, 2005, p354).
- b. Tipe Entitas Lemah, yaitu tipe entitas yang keberadaannya bergantung pada tipe entitas lainnya (Connolly, 2005, p355).



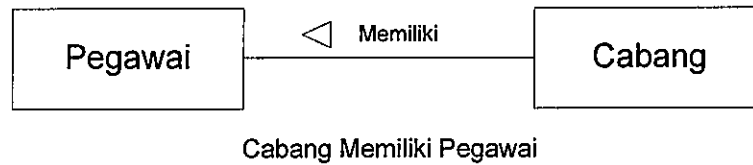
**Gambar 2.4 Contoh Representasi Diagram Tipe Entitas Kuat dan Entitas Lemah (Connolly and Begg 2005, p355)**

#### 2.1.6.2 Relationship Types

Tipe relasi adalah sekumpulan hubungan antar tipe entitas yang memiliki arti (Connolly, 2005, p346). Sekumpulan *relationship occurrence* adalah sebuah hubungan yang dapat diidentifikasi secara unik, yang meliputi sebuah kejadian (*occurrence*) dari setiap tipe entitas di dalam relasi (Connolly, 2005, p346).

Tipe relasi digambarkan dengan sebuah garis yang menghubungkan entitas-entitas yang saling berhubungan. Garis tersebut diberi nama sesuai dengan nama hubungannya dan diberi tanda panah satu arah disamping nama hubungannya.

Biasanya sebuah relasi dinamakan dengan menggunakan kata kerja, seperti Mengatur, atau dengan sebuah frame singkat yang meliputi sebuah kata kerja, seperti DisewaOleh, sedangkan tanda panah ditempatkan di bawah nama relasi yang mengindikasikan arah bagi pembaca untuk mengartikan nama dari suatu relasi. Huruf pertama dari setiap kata pada suatu relasi ditulis dengan huruf besar.



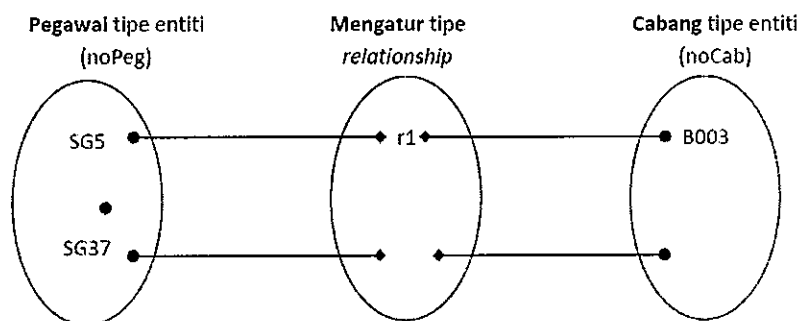
**Gambar 2.5 Contoh Representasi Diagram dari Tipe Relasi**

(Connolly and Begg 2005,p347)

Tipe-tipe relasi yaitu :

a. *One-to-one Relationship*

Gambar di bawah ini menggambarkan *relationship one-to-one* antara entitas *Staff* dan entitas *Branch*, dimana satu orang *staff* hanya mengontrol satu cabang dan satu cabang hanya dikontrol oleh satu orang *staff*.

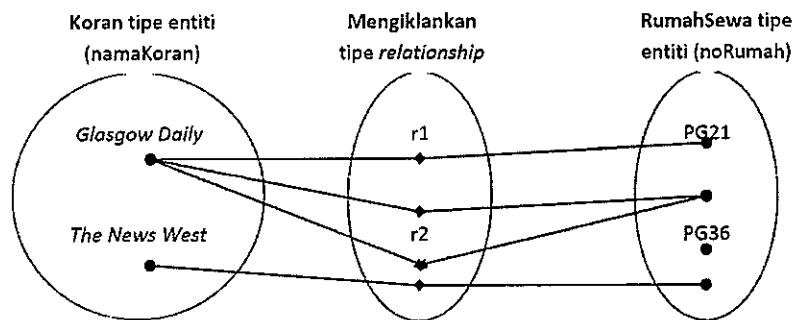


**Gambar 2.6 Contoh Semantic Net Menunjukkan Dua Occurence dari**

**Relasi Pegawai Mengatur Cabang (Connolly and Begg 2005, p357)**

c. *Many-to-many Relationship*

Gambar di bawah ini menggambarkan *relationship many-to-many* yang terjadi diantara entitas *Newspaper* dengan entitas *Property*, dimana satu *property* dapat dipasarkan (*advertise*) pada lebih dari satu koran, begitu juga satu buah koran dapat memasarkan lebih dari satu *property*.



**Gambar 2.8 Contoh Semantic net menunjukkan empat occurrence dari relasi Koran Mengiklankan RumahSewa (Sumber : Connolly and Begg 2005, p360)**

### 2.1.6.3 Attributes

Adapun tipe-tipe atribut sebagai berikut:

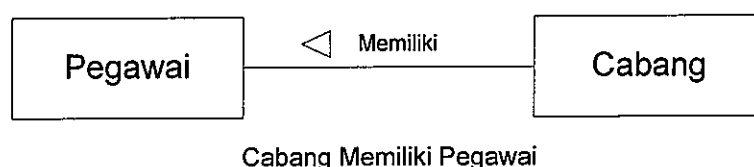
a. *Single valued attribute*

Atribut yang hanya memiliki sebuah nilai untuk setiap *occurrence* dari sebuah entitas (Connolly, 2005, p351).

b. *Multi valued attribute*

Atribut yang memiliki banyak nilai untuk setiap *occurrence* dari sebuah entitas (Connolly, 2005, p352).

Biasanya sebuah relasi dinamakan dengan menggunakan kata kerja, seperti Mengatur, atau dengan sebuah frame singkat yang meliputi sebuah kata kerja, seperti DisewaOleh, sedangkan tanda panah ditempatkan di bawah nama relasi yang mengindikasikan arah bagi pembaca untuk mengartikan nama dari suatu relasi. Huruf pertama dari setiap kata pada suatu relasi ditulis dengan huruf besar.



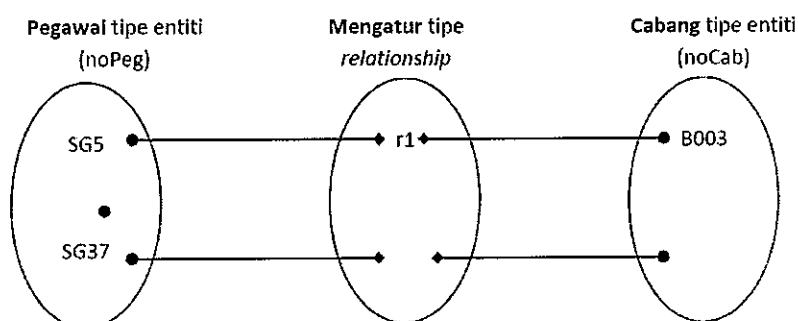
**Gambar 2.5 Contoh Representasi Diagram dari Tipe Relasi**

(Connolly and Begg 2005,p347)

Tipe-tipe relasi yaitu :

a. *One-to-one Relationship*

Gambar di bawah ini menggambarkan *relationship one-to-one* antara entitas *Staff* dan entitas *Branch*, dimana satu orang *staff* hanya mengontrol satu cabang dan satu cabang hanya dikontrol oleh satu orang *staff*.

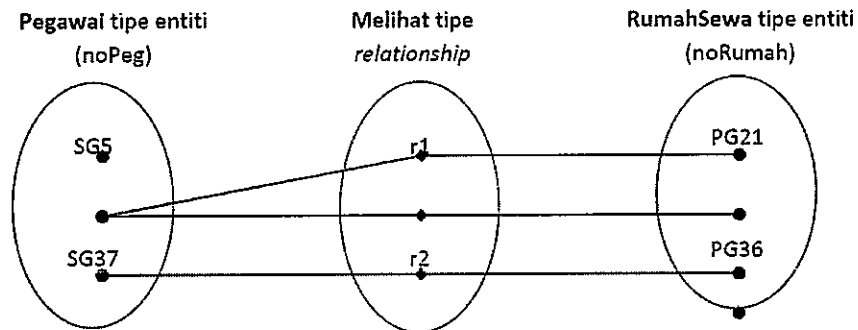


**Gambar 2.6 Contoh Semantic Net Menunjukkan Dua Occurrence dari**

**Relasi Pegawai Mengatur Cabang (Connolly and Begg 2005, p357)**

b. *One-to-many Relationship*

Gambar berikut ini menggambarkan *relationship one-to-many* yang sering terjadi antara entitas *Staff* dengan entitas Properti, dimana dalam relasi ini seorang *staff* memungkinkan untuk mengurus (*oversees*) lebih dari satu properti, sedangkan satu properti hanya dapat diurus oleh satu *staff*.

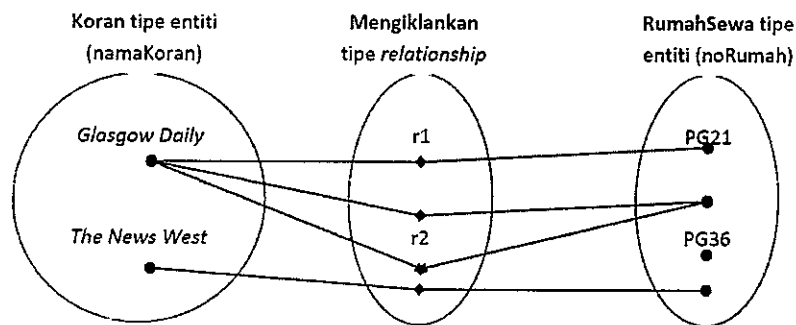


**Gambar 2.7 Contoh Semantic Net Menunjukkan tiga occurrence dari relasi**

**Pegawai Melihat Rumah Sewa (Sumber : Connolly and Begg 2005, p358)**

c. *Many-to-many Relationship*

Gambar di bawah ini menggambarkan *relationship many-to-many* yang terjadi diantara entitas *Newspaper* dengan entitas *Property*, dimana satu *property* dapat dipasarkan (*advertise*) pada lebih dari satu koran, begitu juga satu buah koran dapat memasarkan lebih dari satu *property*.



**Gambar 2.8 Contoh Semantic net menunjukkan empat occurrence dari relasi Koran Mengiklankan RumahSewa (Sumber : Connolly and Begg 2005, p360)**

### 2.1.6.3 Attributes

Adapun tipe-tipe atribut sebagai berikut:

a. *Single valued attribute*

Atribut yang hanya memiliki sebuah nilai untuk setiap *occurrence* dari sebuah entitas (Connolly, 2005, p351).

b. *Multi valued attribute*

Atribut yang memiliki banyak nilai untuk setiap *occurrence* dari sebuah entitas (Connolly, 2005, p352).



c. *Derived attribute*

Atribut yang nilai-nilainya diperoleh dari pengolahan atau diturunkan dari atribut lain yang berhubungan (Connolly, 2005, p352).

#### 2.1.6.4 Keys

*Candidate key* adalah himpunan atribut minimal yang secara unik mengidentifikasi setiap *occurrence* dari sebuah tipe entitas (Connolly, 2005, 352).

*Composite key* adalah sebuah *candidate key* yang terdiri atas dua atau lebih atribut (Connolly, 2005, p353).

*Primary key* adalah *candidate key* yang terpilih untuk mengidentifikasi secara unik setiap *occurrence* dari sebuah tipe entitas (Connolly, 2005, p353).

Pada sebuah tipe entitas biasanya terdapat lebih dari satu *candidate key* yang salah satunya harus dipilih untuk menjadi *primary key*.

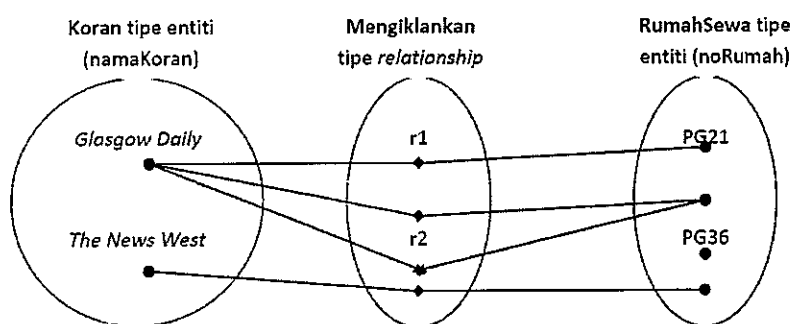
Pemilihan *primary key* didasarkan pada panjang atribut, jumlah minimal atribut yang diperlukan dan keunikannya.

*Alternate key* adalah setiap *candidate key* yang tidak terpilih menjadi *primary key*, atau biasa disebut dengan *secondary key* (Connolly, 2005, p79).

*Foreign key* adalah sebuah *primary key* pada sebuah entitas yang digunakan pada entitas lainnya untuk mengidentifikasi sebuah *relationship* (Connolly, 2005, p79).

c. *Many-to-many Relationship*

Gambar di bawah ini menggambarkan *relationship many-to-many* yang terjadi diantara entitas *Newspaper* dengan entitas *Property*, dimana satu *property* dapat dipasarkan (*advertise*) pada lebih dari satu koran, begitu juga satu buah koran dapat memasarkan lebih dari satu *property*.



**Gambar 2.8 Contoh Semantic net menunjukkan empat occurrence dari relasi Koran Mengiklankan RumahSewa (Sumber : Connolly and Begg 2005, p360)**

### 2.1.6.3 Attributes

Adapun tipe-tipe atribut sebagai berikut:

a. *Single valued attribute*

Atribut yang hanya memiliki sebuah nilai untuk setiap *occurrence* dari sebuah entitas (Connolly, 2005, p351).

b. *Multi valued attribute*

Atribut yang memiliki banyak nilai untuk setiap *occurrence* dari sebuah entitas (Connolly, 2005, p352).

c. *Derived attribute*

Atribut yang nilai-nilainya diperoleh dari pengolahan atau diturunkan dari atribut lain yang berhubungan (Connolly, 2005, p352).

#### 2.1.6.4 Keys

*Candidate key* adalah himpunan atribut minimal yang secara unik mengidentifikasi setiap *occurrence* dari sebuah tipe entitas (Connolly, 2005, 352).

*Composite key* adalah sebuah *candidate key* yang terdiri atas dua atau lebih atribut (Connolly, 2005, p353).

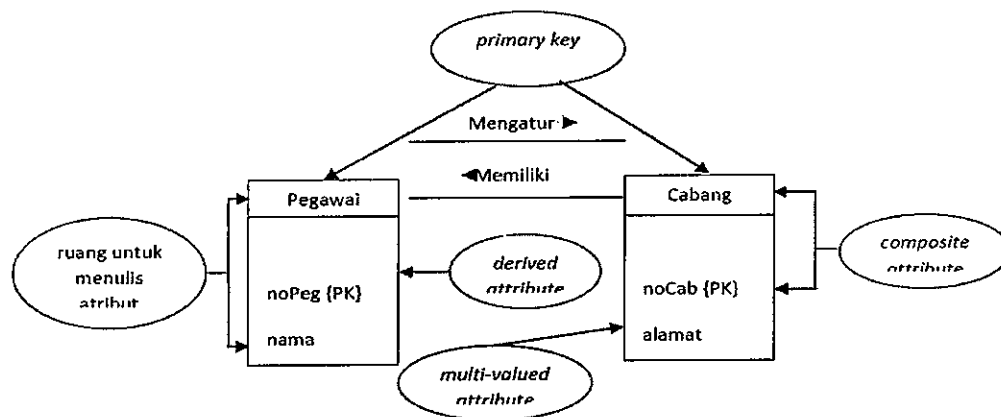
*Primary key* adalah *candidate key* yang terpilih untuk mengidentifikasi secara unik setiap *occurrence* dari sebuah tipe entitas (Connolly, 2005, p353).

Pada sebuah tipe entitas biasanya terdapat lebih dari satu *candidate key* yang salah satunya harus dipilih untuk menjadi *primary key*.

Pemilihan *primary key* didasarkan pada panjang atribut, jumlah minimal atribut yang diperlukan dan keunikannya.

*Alternate key* adalah setiap *candidate key* yang tidak terpilih menjadi *primary key*, atau biasa disebut dengan *secondary key* (Connolly, 2005, p79).

*Foreign key* adalah sebuah *primary key* pada sebuah entitas yang digunakan pada entitas lainnya untuk mengidentifikasi sebuah *relationship* (Connolly, 2005, p79).



**Gambar 2.9 Representasi Diagram Entitas Pegawai dan Cabang Beserta Atribut dan Primary Key-nya (Connolly and Begg 2005, p354)**

#### 2.1.6.5 Structural Constraint

Menurut Connolly (2005,p356), batasan utama pada *relationship* disebut *multiplicity*, yaitu jumlah (atau *range*) dari kejadian yang mungkin terjadi pada suatu entitas yang terhubung ke satu kejadian dari entitas lain yang berhubungan melalui suatu *relationship*. *Relationship* yang paling umum adalah *binary relationship*. Macam – macam *binary relationship*:

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

## 2.1.7 Metodologi Perancangan Basisdata

### 2.1.7.1. Perancangan Basis Data Konseptual

Menurut Connolly (2005,p442), perancangan basis data konseptual adalah proses membangun model informasi yang digunakan dalam perusahaan, terlepas dari segala pertimbangan fisik.

Langkah-langkah perancangan basis data konseptual adalah sebagai berikut :

#### **Langkah 1 : Membangun model data konseptual**

Tujuan dari langkah ini adalah untuk membangun model data konseptual terhadap kebutuhan data suatu perusahaan.

##### **Langkah 1.1 : mengidentifikasi tipe entity**

Pada tahapan ini dilakukan identifikasi tipe *entity* utama yang diperlukan oleh *view*. Salah satu metode dalam mengidentifikasi tipe *entity* adalah dengan memeriksa kebutuhan *user*.

##### **Langkah 1.2 : Mengidentifikasi tipe *relationship***

Pada tahapan ini dilakukan identifikasi relasi penting yang ada antara tipe *entity* yang telah teridentifikasi. Ketika mengidentifikasi relasi kita dapat menggunakan kata kerja dalam spesifikasi kebutuhan *user*.

##### **Langkah 1.3 : Mengidentifikasi dan mengasosiasikan *attributes* dengan *entity* atau tipe *relationship***

Pada tahapan ini, atribut dihubungkan dengan *entity* atau tipe *entity* yang tepat. *Attribute* menyimpan nilai dari setiap *entity* dan mewakili bagian utama dari data yang disimpan dalam basis data.

Sebagai contoh, tipe *entity Staff* dapat dideskripsikan oleh *attribute staffNo, name, position, dan salary*.

#### **Langkah 1.4 : Menentukan Domain Attribute**

Pada tahapan ini, dilakukan penentuan *domain* dari setiap *attribute* dalam model data dan mendokumentasikan detail dari setiap *domain*. *Domain* mendefenisikan nilai-nilai yang dimiliki sebuah *attribute* dan sama dengan konsep *domain* pada *relational model*.

#### **Langkah 1.5 : Menentukan *candidate key, primary key dan alternate key* tiap atribut**

Tujuan dari langkah ini adalah untuk mengidentifikasi *candidate key* dari setiap tipe entitas, dan jika memang terdapat lebih dari satu *candidate key*, pilihlah salah satunya untuk menjadi *primary key*, dan yang lainnya sebagai *alternate key*.

#### **Langkah 1.6 : Mempertimbangkan konsep *enhanced modeling (optional)***

Tahap ini bersifat *optional*, apakah akan digunakan pengembangan dari entity model dengan menggunakan spesialisasi, generalisasi, agregasi dan komposisi yang berguna untuk mendukung *enhanced modeling*.

#### **Langkah 1.7 : Mengecek *redundancy***

Pada tahapan ini, bertujuan untuk memeriksa model data konseptual local apakah terjadi duplikasi atau tidak. Adapun langkah-langkahnya sebagai berikut :

- Memeriksa kembali *one to one (1:1) relationship*

Dalam mengidentifikasi *entity-entity*, ada dua *entity* yang merepresentasikan hal yang sama dalam perusahaan.

- Menghilangkan *redundancy* pada *relationship* yang ada  
Sebuah *relationship* dikatakan *redundancy*, apabila informasi yang sama dapat diperoleh dari *relationship* yang lain.

#### **Langkah 1.8 : Memvalidasi model konseptual lokal dengan transaksi *user***

Memeriksa model yang telah dihasilkan apakah mendukung transaksi pada *view*. Pemeriksaan ini dapat menggunakan dua langkah, yaitu :

- Mendeskripsikan transaksi
- Menggunakan jalur transaksi

#### **Langkah 1.9 : *Mereview* model data konseptual dengan user**

Bertujuan untuk *mereview* model data konseptual dengan *user* untuk memastikan apakah data model sudah benar dengan mempertimbangkan representasi kebutuhan data perusahaan yang sesuai.

### 2.1.7.2 Perancangan Basis Data Logikal

Menurut Connolly (2005, p294), perancangan basis data logical adalah proses membangun model informasi yang digunakan dalam perusahaan dengan berdasar pada suatu model data spesifik tetapi masih terlepas dari DBMS tertentu beserta pertimbangan fisik lainnya.

Menurut Connolly (2005, p462), langkah-langkah perancangan basis data logikal adalah sebagai berikut :

#### **Langkah 2 : Membangun dan memvalidasi model data logical untuk setiap view**

Tujuan dari langkah ini adalah untuk menerjemahkan model data konseptual kedalam model data logikal dan kemudian memvalidasi model tersebut untuk mengecek apakah secara struktur benar dan mendukung transaksi yang dibutuhkan.

#### **Langkah 2.1 : Menentukan relasi untuk model logikal data lokal**

Tujuan dari langkah ini adalah untuk membuat suatu relasi untuk model logical data local yang merepresentasikan suatu *entity*, relasi, dan juga atribut yang telah diidentifikasi. Adapun pendeskripsian bagaimana relasi dapat diturunkan dari struktur data model yang ada, antara lain :

- Tipe *strong entity*
- Tipe *weak entity*
- Tipe relasi *binary one-to-many* (1 : \*)



- Tipe relasi *binary one-to-one* (1 : 1)
- Relasi rekursif *one-to-one* (1 : 1)
- Tipe relasi *superclass / subclass*
- Tipe relasi *binary many-to-many*
- Tipe relasi kompleks
- Atribut *multi-valued*

### **Langkah 2.2 : Memvalidasi relasi menggunakan normalisasi**

Normalisasi digunakan untuk meningkatkan model yang telah terbentuk agar duplikasi data yang tidak diperlukan dapat dihindari.

### **Langkah 2.3 : Memvalidasi relasi terhadap transaksi-transaksi *user***

Langkah ini dilakukan dengan tujuan untuk memastikan bahwa relasi dalam model data logikal lokal mendukung transaksi-transaksi yang diperlukan oleh *view*.

### **Langkah 2.4 : Mengecek batasan integritas**

Tujuan dari langkah ini adalah untuk mengecek batasan integritas yang direpresentasikan dalam model data logikal. Batasan integritas merupakan batasan yang diharapkan dapat menjaga basis data agar tidak menjadi tidak lengkap (*incomplete*), tidak akurat (*inaccurate*), atau tidak konsisten (*inconsistent*).

Dibawah ini terdapat enam tipe batasan integritas, antara lain :

- a. Data yang dibutuhkan;
- b. Batasan *domain* atribut;
- c. *Multiplicity*;
- d. Integritas entitas;

- e. Integritas referensial;
- f. Batasan umum.

**Langkah 2.5 : Me-review model data logikal terhadap kebutuhan *user***

Tujuan dari langkah ini adalah untuk me-review model data logikal dengan *user* untuk memastikan bahwa model tersebut sesuai dengan representasi yang benar dari kebutuhan data perusahaan. Apabila *user* merasa tidak puas dengan model tersebut maka dilakukan pengulangan kembali langkah-langkah sebelumnya jika diperlukan.

**Langkah 2.6 : Menggabungkan model data logikal kedalam model global (langkah *optional*)**

Tujuan dari langkah ini adalah untuk menggabungkan suatu model data logikal lokal kedalam satu model data logikal global yang merepresentasikan semua sudut pandang *user* terhadap basis data.

**Langkah 2.7 : Mengecek kemungkinan pengembangan di masa depan**

Tujuan dari langkah ini adalah untuk menentukan bagian mana yang kelihatannya akan berubah ke masa depannya dan juga memperhatikan supaya model data logikal dapat mengakomodasi perubahan tersebut.

### 2.1.7.3 Perancangan Basis Data Fisikal

Menurut Connolly and begg (2005, p294), perancangan basis data fisikal adalah suatu proses untuk mendeskripsikan pengimplementasian dari suatu basis data pada media penyimpanan *secondary*, dengan mendeskripsikan relasi dasar, organisasi file, dan indeks yang digunakan untuk mencapai keefisienan dalam mengakses data, dan batasan integritas, serta pengukuran keamanan apapun yang berhubungan.

#### **Langkah 3 : Menerjemahkan Model Data Logikal sesuai DBMS yang Dituju**

Tujuan dari langkah ini adalah untuk membuat suatu skema basis data relasional dari model data logikal yang dapat diimplementasikan ke DBMS yang dituju.

##### **Langkah 3.1 : Merancang relasi dasar**

Tujuan dari langkah ini adalah untuk memutuskan bagaimana merepresentasikan relasi dasar yang diidentifikasi dalam model data logikal pada DBMS yang dituju.

Untuk memulai proses perancangan basis data fisikal, pertama-tama dapat dilakukan dengan menyatukan dan mengasimilasikan informasi mengenai relasi yang dirancang selama perancangan basis data logikal. Informasi yang diperlukan dapat berasal dari kamus data dan definisi relasi yang didefinisikan menggunakan

**Langkah 3.2 : Merancang representasi data turunan (*derived data*)**

Tujuan dari langkah ini adalah untuk memutuskan bagaimana merepresentasikan suatu data turunan yang terdapat pada model data logikal pada DBMS yang dituju.

Atribut yang nilainya didapatkan dengan mengevaluasi atribut lain dikenal sebagai atribut turunan atau atribut kalkulasi. Sebagai contoh :

- a. Jumlah staf yang bekerja pada suatu cabang (*branch*).
- b. Total gaji bulanan untuk semua staf.
- c. Jumlah properti yang di-*handle* oleh anggota staf.

**Langkah 3.3 : Merancang *general constraint***

Tujuan dari langkah ini adalah untuk merancang kendala umum untuk DBMS yang dituju. Meng-*update* suatu relasi yang mungkin dibatasi oleh batasan integritas yang mengatur transaksi 'real world' yang direpresentasikan oleh peng-*update*-an tersebut. Perancangan batasan tersebut sekali lagi tergantung pada DBMS yang dipilih. Beberapa sistem menyediakan fasilitas-fasilitas dibandingkan yang lainnya untuk mendefinisikan kendala umum. Seperti langkah sebelumnya, jika sistem tersebut mempunyai aturan sesuai aturan standar SQL, beberapa batasan dapat diterapkan.

#### **Langkah 4 : Merancang Organisasi *File* dan Indeks**

Tujuan dari langkah ini adalah untuk menentukan organisasi *file* yang optimal untuk menyimpan relasi dasar dan indeks yang dibutuhkan untuk mencapai kinerja yang diharapkan. Karena itu, cara dimana relasi dan *tuples* yang ada akan disimpan pada penyimpanan *secondary*.

##### **Langkah 4.1 : Menganalisis transaksi**

Tujuan dari langkah ini adalah untuk memahami fungsionalitas dari suatu transaksi dimana akan dijalankan pada basis data untuk menganalisis transaksi yang penting.

##### **Langkah 4.2 : Memilih organisasi *file***

Tujuan dari langkah ini adalah untuk menentukan organisasi *file* yang efisien untuk setiap relasi dasar. Beberapa organisasi *file* efisien untuk *bulk loading data* kedalam basis data tetapi setelah itu tidak efisien. Dengan kata lain, kita ingin menggunakan struktur penyimpanan yang efisien untuk mengeset basis data dan kemudian mengubahnya untuk penggunaan operasional normal.

Karena itu, tujuan dari langkah ini adalah untuk memilih organisasi *file* yang optimal untuk tiap relasi, jika DBMS yang dituju memperbolehkannya. Dalam banyak kasus yang ada, suatu relasional DBMS akan memberikan sedikit bahkan tanpa pilihan dalam memilih organisasi *file*, walaupun beberapa akan mempunyai indeks yang spesifik

### **Langkah 4.3 : Memilih indeks**

Tujuan dari langkah ini adalah untuk menentukan apakah dengan adanya penambahan indeks akan meningkatkan kinerja dari suatu sistem.

Salah satu pendekatan untuk memilih organisasi *file* yang sesuai untuk relasi yaitu menjaga agar *tuple* tidak berurutan dan membuat indeks *secondary* sebanyak mungkin. Pendekatan lainnya yaitu untuk mengurutkan *tuple* dalam relasi dengan menspesifikasi *primary* atau *clustering* indeks.

### **Langkah 4.4 : Mengestimasi kapasitas penyimpanan**

Tujuan dari langkah ini adalah untuk mengestimasi jumlah kapasitas *disk* yang akan dibutuhkan oleh basis data dalam mendukung implementasi basis data pada penyimpanan *secondary*.

Seperti pada langkah sebelumnya, mengestimasi penggunaan *disk* sangat bergantung pada DBMS yang dituju dan perangkat lunak yang digunakan untuk mendukung basis data. Secara umum estimasi tersebut dilakukan berdasarkan ukuran tiap *tuple* dan jumlah *tuple* dalam relasi.

### **Langkah 5 : Merancang Tampilan untuk User**

Tujuan dari langkah ini adalah untuk merancang tampilan *user* yang diidentifikasi selama tahap pengumpulan dan analisis kebutuhan pada Siklus Hidup Pengembangan Sistem Basis Data.

### **Langkah 6 : Merancang Mekanisme Keamanan**

Tujuan dari langkah ini adalah untuk merancang mekanisme keamanan untuk basis data seperti yang telah dispesifikasikan *user* selama tahap analisis dan pengumpulan kebutuhan pada Siklus Hidup Pengembangan Sistem Basis Data.

Selama tahap analisis dan pengumpulan kebutuhan pada Siklus Hidup Pengembangan Sistem Basis Data, kebutuhan keamanan yang spesifik harus didokumentasikan dalam spesifikasi kebutuhan sistem. Sasaran dari tahap ini adalah untuk memutuskan bagaimana kebutuhan keamanan ini akan direalisasikan. Beberapa sistem menawarkan fasilitas keamanan yang berbeda dari sistem yang lainnya. Sekali lagi, perancang basis data harus hati-hati terhadap fasilitas yang ditawarkan oleh DBMS yang dituju. Relasional DBMS biasanya menyediakan dua macam keamanan basis data antara lain :

- a. Keamanan sistem
- b. Keamanan data

Keamanan sistem menutupi pengaksesan dan penggunaan basis data pada tingkat sistem, seperti *username* dan *password*. Sedangkan keamanan data menutupi pengaksesan dan penggunaan objek basis data (seperti relasi dan *views*) dan tindakan dimana *user* dapat memperoleh objek tersebut.

Definisi dari keamanan basis data adalah suatu mekanisme yang memproteksi basis data dari suatu kejadian baik yang disengaja maupun tidak. Suatu basis data merupakan sumber dari perusahaan yang *essential* yang perlu dilindungi dengan menggunakan suatu kontrol yang memadai. Beberapa *issue* keamanan yang perlu diperhatikan :

- a. Pencurian data (*Theft and Fraud*)
- b. Kehilangan kerahasiaan data (*Loss of Confidentially*)
- c. Kehilangan hak pribadi (*Loss of Privacy*)

- d. Kehilangan integritas (*Loss of integrity*)
- e. Kehilangan ketersediaan data (*Loss of availability*)

Hasil akhir perancangan fisik basis data adalah suatu proses yang mendeskripsikan suatu implementasi dari suatu basis data pada media penyimpanan. Hal ini mendeskripsikan suatu relational dan struktur penyimpanan dan metodologi pengaksesan data oleh *user* yang efisien, selama batasan integritas dan pengukuran keamanan.

### 2.1.8 Normalisasi

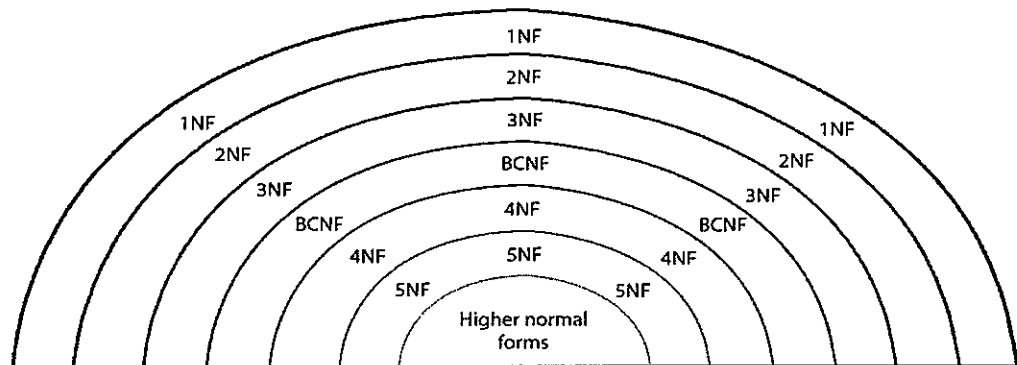
Normalisasi adalah sebuah teknik untuk menghasilkan sejumlah relasi dengan sifat-sifat (*properties*) yang diinginkan berdasarkan kebutuhan-kebutuhan data suatu perusahaan untuk memenuhi kebutuhan data pada perusahaan/organisasi (Connolly, 2005, p388)

Tujuan dari normalisasi adalah untuk menghilangkan kerangkapan data, untuk mengurangi kompleksitas dan untuk mempermudah memodifikasi data.

Empat bentuk normal yang biasa digunakan yaitu, *first normal form* (1NF), *second normal form* (2NF), *third normal form* (3NF) dan Boyce-Codd *normal form* (BCNF). Konsep utamanya terkait dengan *functional dependencies*, dimana menerangkan hubungan antar atribut yang ada. Sebuah relasi dapat dinormalisasikan ke dalam bentuk tertentu untuk mengatasi kemungkinan terjadinya pengulangan dari update yang tidak baik. (Connolly, 2005, p388)



Hubungan antara normal *forms* menurut Connolly (2005, p401-422) sebagai berikut:



**Gambar 2.10 Diagram Ilustrasi dari Hubungan di Antara Normal Forms (Connolly, 2005, p401)**

#### 1. *Unnormalized Form* (UNF)

Menurut Connolly (2005,p403), pada bentuk tidak normal (*Unnormalized Form – UNF*), table masih mengandung satu atau lebih kelompok perulangan (*repeating groups*). Tabel UNF ini dibuat dengan mentransformasikan data dari sebuah informasi ke dalam table berbentuk baris dan kolom.

#### 2. *First Normal Form* (1NF)

Menurut Connolly (2005,p403), pada bentuk normal pertama (*First Normal Form – 1NF*), suatu relasi dimana pada setiap sel (perpotongan dari baris dan kolom) memuat satu dan hanya satu nilai, setiap sel mengandung nilai *atomic* (atau *single value*).

### 3. *Second Normal Form (2NF)*

Menurut Connolly (2005,p407), pada bentuk normal kedua (*Second Normal Form – 2NF*), suatu relasi telah melalui bentuk pertama dan setiap atribut bukan *primary key* tergantung fungsional terhadap *primary key*.

### 4. *Third Normal Form (3NF)*

Menurut Connolly (2005,p409), pada bentuk normal ketiga (*Third Normal Form – 3NF*), suatu relasi telah melalui bentuk pertama dan kedua, serta tidak ada attribute bukan *primary key* yang tergantung secara transitif kepada *primary key*.

### 5. *Boyce-Codd Normal Form (BCNF)*

Relasi yang disebut BCNF, adalah relasi yang hanya dan hanya jika setiap determinan adalah *candidate key*.

### 6. *Fourth Normal Form (4NF)*

4NF yaitu relasi yang terdapat dalam *Boyce Codd Normal Form* dan tidak mengandung *multi valued dependencies*.

## 2.2 Sistem Berbasis Web

### 2.2.1 Intranet

Menurut Khoe Yao Tung(1997,p4) Intranet adalah jaringan komputer dalam perusahaan yang menggunakan komunikasi data standar seperti dalam internet .Artinya,kita dapat menggunakan semua fasilitas internet untuk kebutuhan dalam perusahaan.Dengan kata lain intranet dapat dikatakan berinternet dalam lingkungan perusahaan.

Menggunakan standar protokol TCP/IP,TCP/IP memungkinkan protokol jaringan untuk berkomunikasi,menerima dan mengirimkan data ke terminal yang lain.Standar yang lain adalah FTP (File Transfer Protocol) yang merupakan pelayanan resource sharing,sebuah fasilitas untuk dapat mengambil file yang ada di internet.SMTP yang merupakan dasar dari e-mail untuk berkomunikasi serta MIME yang merupakan standar untuk mendefinisikan format biner,grafik dan suara agar dapat ditransmisikan dengan e-mail.Protokol lainnya adalah NNTP(Network News Transfer Protocol) dan POP (Post Office Protocol).

Secara umum teknologi yang digunakan antara Internet dan Intranet adalah sama.Namun demikian terdapat perbedaan antara Intranet dan Internet dilihat dari perspektif jangkauan dan lingkup penggunaanya.

- Lingkup akses dan jangkauan.
- Cara teknologi yang digunakan untuk berkomunikasi.
- Tujuan dari terselenggaranya komunikasi.

### 2.2.2 Web

Menurut Raymond Mcleod,Jr.,George P.Schell(2007,p78) World Wide Web, yang disebut pula sebagai Web dan WWW,adalah informasi yang dapat diakses melalui internet di mana dokumen-dokumen hypermedia(file-file komputer) disimpan dan kemudian diambil dengan cara-cara yang menggunakan metode penentuan alamat yang unik. World Wide Web adalah kumpulan komputer yang bertindak sebagai server isi/kandungan yang menyimpan dokumen-dokumen yang diformat untuk memungkinkan dilihatnya teks,grafik,dan audio.sekaligus juga memungkinkan dibuatnya link dengan dokumen-dokumen lain di dalam Web. Server-server dan para pengguna yang mengaksesnya ini tersambung oleh internet.

### 2.2.3 HTTP

HTTP merupakan singkatan dari Hypertext Transfer Protocol.Jika diartikan secara bebas, HTTP berarti sebuah protokol (atau bahasa) yang digunakan untuk menyampaikan informasi di internet.

HTTP juga merupakan bahasa yang digunakan oleh client dan server untuk saling berkomunikasi.Penggunaan protokol ini juga seringkali erat hubungannya dengan nama domain atau nama situs.

## 2.2.4 Lingkungan Programming

### 1. PHP

Menurut Ali Zaki & SmitDev Community (2008,p2) PHP adalah sebuah bahasa pemrograman scripting untuk membuat halaman web yang dinamis,Walaupun dikenal sebagai bahasa untuk membuat halaman web,tapi php sebenarnya juga dapat digunakan untuk aplikasi command line dan juga GUI.Cara kerja PHP adalah dengan menyelipkannya diantara kode HTML (Hypertext markup language).

Website yang dibuat menggunakan PHP memerlukan software bernama webserver tempat pemrosesan kode PHP dilakukan.PHP bersifat terbuka dan multiplatform,karenanya dapat dijalankan di banyak merek web server (seperti Apache dan IIS).

### 2. MySQL

Menurut Stendy B.Sakur (2005,p57) MySQL merupakan salah satu system database yang sangat handal karena menggunakan system SQL.Pada awalnya SQL berfungsi sebagai bahasa penghubung antara program database dengan bahasa pemograman yang kita gunakan.Setelah itu SQL dikembangkan lagi menjadi system database dengan munculnya MySQL.

MySQL merupakan database yang sangat cepat, beberapa user dapat menggunakan secara bersamaan , dan lebih lengkap dari SQL. MySQL merupakan software gratis yang dapat di-download melalui situsnya. MySQL merupakan sistem manajemen database, relasional sistem database dan software open source.

### **3. XAMPP**

Menurut Yogi Wicaksono (2008,p7), XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya.

## 2.3 Tools yang digunakan untuk Analisis dan Perancangan

### 2.3.1. Data Flow Diagram (DFD)

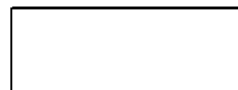
Menurut McLeod (2001,p401), *Data Flow Diagram* (DFD) adalah representasi grafis dari sistem yang menggunakan sejumlah simbol-simbol untuk mengilustrasikan bagaimana aliran data melewati proses-proses yang saling berhubungan.

Gambar aliran data sebagai *modeling tool* dengan menggunakan pendekatan metode analisis sistem terstruktur. DFD dapat digunakan untuk mempresentasikan suatu sistem yang otomatis maupun manual dengan DFD menggunakan simbol :

#### 1. Entitas Eksternal (Terminal)

Entitas yang berada diluar sistem , yang memberikan data kepada sistem (*source*) atau yang menerima informasi dari sistem (*sink*). Entitas eksternal tidak termasuk dalam sistem.

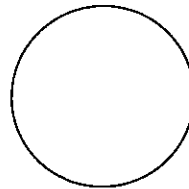
Simbol yang digunakan :



## 2. Proses

Menggambarkan apa yang dilakukan oleh sistem. Proses berfungsi untuk mentransformasikan suatu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang digunakan. Setiap proses memiliki satu atau beberapa data masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

Simbol yang digunakan :



## 3. Data Flow

Menggambarkan aliran data dari suatu entitas ke entitas lainnya. Tanda panah menggambarkan aliran data yang terjadi bisa antara dua proses yang berurutan, dari data store ke process atau sebaliknya dari process ke sink.

Simbol yang digunakan :



## 4. Data Store

Data store berfungsi untuk sebagai tempat penyimpanan data. Yaitu proses dapat mengambil data atau memberikan data ke data store.



Simbol yang digunakan :





### 2.3.2. State Transition Diagram ( STD)

Menurut Whitten, Bentley, dan Dittman (2004, p673), *state transition diagram* adalah sebuah alat yang digunakan untuk menggambarkan urutan dan variasi layar yang dapat terjadi selama sesi yang dapat digunakan oleh *user*.

Simbol	Nama	Keterangan
	<i>State</i>	Merupakan kumpulan keadaan atau atribut yang mencirikan seseorang atau benda pada waktu atau kondisi tertentu.
<p>“klik keluar”</p> <hr/> <p>Keluar dari aplikasi</p> 	<i>Transition State</i>	Merupakan perubahan yang digambarkan dengan simbol panah. Setiap panah diberi label

### 2.3.3. Entity Relationship Diagram

Pada dasarnya ERD (Entity Relationship Diagram) adalah sebuah diagram yang secara konseptual memetakan hubungan antara penyimpanan pada diagram DFD.

ERD ini dilakukan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah objek yang dapat dibedakan dengan objek lainnya. Objek tersebut dapat memiliki komponen-komponen data (atribut atau field) yang membuatnya dapat dibedakan dari objek yang lain.

Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang dikenal dalam entity yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif.

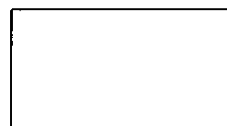
Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut.

Simbol – simbol ERD :

#### 1. Entitas

Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.

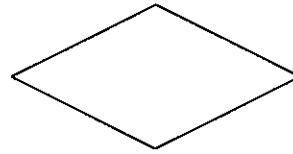
Simbol yang digunakan adalah :



## 2. Relasi

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.

Simbol yang digunakan adalah :



## 3. Garis

Garis sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

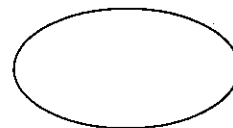
Simbol yang digunakan adalah :



## 4. Atribut

Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

Simbol yang digunakan adalah :



## 2.4 Pemahaman Objek Studi

### Sistem Kekaryawanan

Menurut Henry Simamora (2001 :90), sistem kekaryawanan adalah prosedur sistematis untuk mengumpulkan, menyimpan, memperthankan, menarik, dan memvalidasi data yang dibutuhkan oleh sebuah organisasi tentang sumber daya manusia, aktivitas-aktivitas personalia, karekteristik-karakteristik unit-unit organisasi.

Kegiatan yang berhubungan dengan sumber daya manusia yaitu *Human Resources Information System* (HRIS) sebagai pendukung manajemen sumber daya manusia. HRIS merupakan sebuah sistem yang digunakan untuk memperoleh (*acquire*), menyimpan (*store*), memanipulasi (*manipulate*), menganalisis (*analyze*), mendapatkan kembali (*retrieve*) dan mendistribusikan (*distribute*) information yang berhubungan dengan sumber daya manusia untuk kepentingan organisasi.

### Karyawan

Karyawan adalah orang yang membantu perusahaan untuk mencapai tujuan. Karyawan adalah *asset* perusahaan yang harus dijaga dan dirawat agar tetap dapat menghasilkan keuntungan secara optimal terhadap perusahaan. Dapat diibaratkan

seperti mesin pabrik, jika tidak dirawat (services, ganti oli, ganti spare part) jelas sudah tidak dapat menghasilkan produk secara optimal.

Pada umumnya :

Perusahaan menggunakan karyawan dengan tujuan dapat mendatangkan keuntungan yang sebesar-besarnya bagi perusahaan dengan biaya seminimal mungkin.

Karyawan bekerja pada perusahaan dengan tujuan mendapatkan imbalan yang sebesar-besarnya dengan mengeluarkan tenaga seminimal mungkin.

### **Biodata Karyawan**

Biodata adalah Informasi yang berisi data data mengenai seseorang atau sesuatu mulai dari nama, alamat dan banyak lagi informasi dasar yang dianggap perlu dapat dicantumkan dalam biodata.

Biodata saat ini telah menjadi hal penting dan dibutuhkan dimana saja untuk mempermudah mengenal seseorang. Berikut ini adalah isi dari biodata karyawan:

1. NIK (Nomor Induk Karyawan)
2. Nama Karyawan (berisi nama lengkap karyawan)
3. Jenis Kelamin (berisi laki-laki atau perempuan)
4. Tempat Lahir (berisi nama tempat dilahirkan)
5. Tanggal Lahir

6. Alamat
7. Agama
8. Pendidikan
9. Pendidikan mandiri
10. Band position
11. Loker
12. Jabatan
13. Handphone
14. Telepon
15. E-mail
16. foto

### **Pelatihan**

Pelatihan adalah keseluruhan kegiatan untuk memberi, memperoleh, meningkatkan, serta mengembangkan kompetensi kerja, produktivitas, disiplin, sikap, dan etos kerja pada tingkat keterampilan dan keahlian tertentu sesuai dengan jenjang dan kualifikasi jabatan atau pekerjaan.

Berisi informasi mengenai data pelatihan karyawan yang terdiri dari NIK, Tanggal Mulai pelatihan, Tanggal selesai pelatihan, Nama pelatihan, dan Jenis pelatihan.

## Kompetensi

Russel C. Swansburg (1995,p44) Kompetensi adalah kualitas pribadi atau kemampuan untuk melaksanakan tugas yang diperlukan. Pelatihan dan pendidikan memberikan kompetensi yang perlu untuk menghasilkan keluaran. Saat tuntutan kerja berubah, pelatihan dan pendidikan penting untuk mempertahankan kompetensi.

Berisi informasi mengenai data SKI-kompetensi karyawan yang terdiri dari NIK, Tahun, Nilai SKI, dan Nilai Kompetensi.

## Penghargaan

Penghargaan ialah sesuatu yang diberikan pada perorangan atau kelompok jika mereka melakukan suatu keulungan di bidang tertentu. Penghargaan biasanya diberikan dalam bentuk medali, piala, gelar, sertifikat, plaket atau pita. Suatu penghargaan kadang-kadang disertai dengan pemberian hadiah berupa uang seperti Hadiah Nobel untuk kontribusi terhadap masyarakat, dan Hadiah Pulitzer untuk penghargaan bidang literatur. Penghargaan bisa juga diberikan oleh masyarakat karena pencapaian seseorang tanpa hadiah apa-apa.

Berisi informasi mengenai data penghargaan karyawan yang terdiri dari NIK, Nama penghargaan, Tanggal diberikan, Jenis penghargaan, Judul penghargaan, dan Keterangan.

### **Kenaikan Jabatan (Promosi)**

Sementara itu Moekijat (1999:101) mengungkapkan pengertian lain dari promosi jabatan, yaitu: "Promosi adalah kemajuan yang diperoleh sebagai pegawai pada suatu tugas yang lebih baik, lebih baik dimaksud dipandang dari sudut tanggung jawab yang lebih berat martabat atau status yang lebih tinggi, kecakapan yang lebih baik, dan yang terpenting adalah penambahan jumlah gaji atau upah".

Definisi diatas menjelaskan bahwa promosi yang diperoleh oleh seorang karyawan dipandang sebagai suatu kemajuan. Dimana karyawan akan memiliki suatu pencapaian tugas dan pekerjaan yang lebih baik. Karyawan akan menerima tanggung jawab dan tugas yang lebih tinggi. Tetapi keadaan ini akan diimbangi dengan perolehan kompensasi berupa gaji dan upah yang sebanding dengan tanggung jawab dan tugas yang dijalankan.

Tujuan dari promosi antara lain:

1. Untuk memberikan pengakuan, jabatan, dan imbalan jasa yang semakin besar kepada karyawan yang berprestasi kerja tinggi.
2. Dapat menimbulkan kepuasan dan kebanggaan pribadi, status sosial yang lebih tinggi, dan penghasilan yang semakin besar.
3. Untuk merangsang agar karyawan lebih bergairah kerja, disiplin tinggi, dan memperbesar produktifitas kerjanya.
4. Untuk menjamin stabilitas kepegawaian dengan realisasinya promosi kepada karyawan dengan dasar dan pada waktu yang tepat serta penilaian yang jujur.



5. Kesempatan promosi dapat menimbulkan keuntungan berantai (Multiflier effect) dalam perusahaan karena menimbulkan lowongan berantai.
6. Memberi pegawai kesempatan untuk mengembangkan kreatifitas demi keuntungan perusahaan.
7. Sebagai daya dorong dan pemicu kerja yang lebih baik bagi karyawan lain.

### **Mutasi**

Didalam dalam suatu organisasi, mutasi merupakan bagian dari proses kegiatan yang dapat mengembangkan posisi atau status seseorang dalam suatu organisasi. Karena ia merupakan kekuatan yang mampu mengubah posisi karyawan, maka dikatakan bahwa mutasi merupakan salah satu cara yang paling ampuh untuk mengembangkan sumber daya manusia dalam lingkungan organisasi (Saydam, 1996:97).

Tujuan dari Mutasi:

1. Untuk meningkatkan produktivitas kerja karyawan.
2. Untuk meningkatkan keseimbangan antara tenaga kerja dengan komposisi pekerjaan.
3. Untuk memperluas atau menambah pengetahuan karyawan.
4. Untuk menghilangkan rasa bosan/jemu terhadap pekerjaannya.
5. Untuk memberikan perangsangan agar karyawan mau berupaya meningkatkan karier.

6. Menghilangkan rasa bosan terhadap pekerjaan.
7. Melaksanakan hukuman atas pelanggaran pegawai.
8. Memberikan imbalan terhadap pekerjaan pegawai.
9. Sebagai alat pendorong agar semangat kerja meningkat.

### **Pengumuman**

Menurut Agus Trianto (2006, p78) pengumuman adalah pesan atau informasi yang disampaikan kepada publik (umum). Biasanya, pembuat atau pengirim pengumuman adalah lembaga (pemerintah, perusahaan, organisasi, sekolah), panitia (lomba, festival, amal, kegiatan tertentu), atau individu (perorangan). Tujuan pengumuman adalah menyampaikan sesuatu agar diketahui masyarakat namun pengumuman berbeda dengan iklan. Pengumuman hanya menyampaikan pesan atau informasi agar diketahui masyarakat, selain itu, pengumuman berguna untuk kepentingan umum.

### **FAQ**

FAQ (*Frequently Asked Questions*) adalah kumpulan pertanyaan-pertanyaan yang sering ditemukan, tentunya beserta jawabannya. FAQ adalah sumber informasi yang populer karena berisi jawaban atas masalah-masalah yang dijumpai oleh kebanyakan pengguna web.