

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Sistem**

Dalam pembuatan tugas akhir ini penulis membutuhkan teori-teori yang dapat mendukung kemudahan dalam mempelajari maupun merancang sistem aplikasi yang diharapkan dapat digunakan seoptimal mungkin. Penulis berharap dengan adanya sistem yang dibuat, para pengguna lebih mudah dalam bekerja, sehingga dapat membantu dalam menyelesaikan dengan baik dan dapat memperkecil kesalahan yang dilakukan oleh pengguna. Berikut ini adalah teori-teori yang dapat memperkuat penulis Tugas Akhir ini.

##### **2.1.1. Pengertian Sistem**

Pengertian sistem menurut Pratama (2014:7) “Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”.

Menurut Jerry Fitzgerald, et, al dalam Puspitawati (2011:1) “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”

Sedangkan menurut Norman L Enger dalam Sutabri (2012:7) mengemukakan bahwa “Suatu sistem dapat terdiri atas kegiatan-kegiatan yang berhubungan guna mencapai tujuan-tujuan perusahaan seperti pengendalian inventaris atau penjadwalan produksi.

### 2.1.2. Karakteristik Sistem

Model umum sebuah sistem adalah masukan, proses dan keluaran, hal ini merupakan konsep sebuah sistem yang sangat sederhana, sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu mencirikan bahwa hal tersebut biasa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem ( *Component* )

Suatu sistem terdiri dari jumlah komponen - komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan

2. Batasan Sistem ( *Boundary* )

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini

memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem ( *Environment* )

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut, disebut dengan lingkungan luar sistem. Dimana lingkungan luar ini dapat bersifat menguntungkan dan dapat pula merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, dengan demikian lingkungan luar tersebut harus tetap terjaga dan dapat di pelihara. Sedangkan lingkungan luar sistem yang merugikan harus dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem tersebut.

4. Penghubung Sistem ( *Interface* )

Sebagai media yang menghubungkan sistem dengan subsistem yang lainnya disebut dengan penghubung sistem atau *interface* melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem lainnya. Bentuk keluaran dari suatu subsistem akan menjadi masukan untuk subsistem yang lainnya dengan melalui penghubung tersebut.

5. Masukan Sistem ( *Input* )

Energi yang dimasukkan kedalam sistem tersebut masukan sistem, yang dapat berupa pemeliharaan ( *maintenance input* ) dan signal masukan ( *signal input* ).

#### 6. Keluaran Sistem (*output*)

Keluaran adalah hasil dari energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk sub sistem yang lain atau kepada supra sistem.

#### 7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu proses yang akan merubah masukan menjadi keluaran.

#### 8. Sasaran Sistem

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat determinan kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila menggunakan sasaran atau tujuan yang di rencanakan.

### 2.1.3. Klasifikasi Sistem

Menurut Andri Kristanto (2008:5) sistem dapat diklasifikasikan berdasarkan beberapa sudut pandang, diantaranya adalah sebagai berikut:

#### a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang tidak dapat dilihat secara mata biasa dan biasanya ini berupa pemikiran atau ide-ide, sedangkan Sistem Fisik merupakan sistem yang ada secara fisik.

b. Sistem Alamiah dan Sistem Buatan Manusia

Sistem Alamiah merupakan sistem yang terjadi melalui proses alam, yang tidak dibuat oleh manusia. Sedangkan Sistem Buatan Manusia merupakan sistem yang melibatkan hubungan manusia.

c. Sistem terbuka dan Sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sedangkan Sistem terbuka adalah sistem yang berhubungan dengan lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran.

#### 2.1.4. Pengertian Java

Pengertian Java menurut Utomo (2013:1) yaitu, “Java merupakan salah satu bahasa pemrograman yang bersifat multiplatform dengan slogan dari para pengembangnya sehingga aplikasi yang dikembangkan merupakan bahasa java akan dapat dijalankan pada berbagai macam platform atau sistem operasi”.

Hal ini menjadi salah satu solusi dari berbagai macam bahasa pemrograman yang ada di dunia IT saat ini, yang biasanya hanya dapat dijalankan di sistem operasi yang lain.

#### 2.1.5. Pengertian Rekam Medis

Rekam Medis adalah keterangan baik yang tertulis maupun yang terekam tentang identitas, anamnesis penentuan fisik laboratorium, diagnosis segala pelayanan

dan tindakan medis yang diberikan kepada pasien dan pengobatan baik pada pelayanan masyarakat.

Menurut PERMENKES N0.269/Menkes/PER/III/2008 Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan, dan pelayanan lain yang telah diberikan kepada pasien.

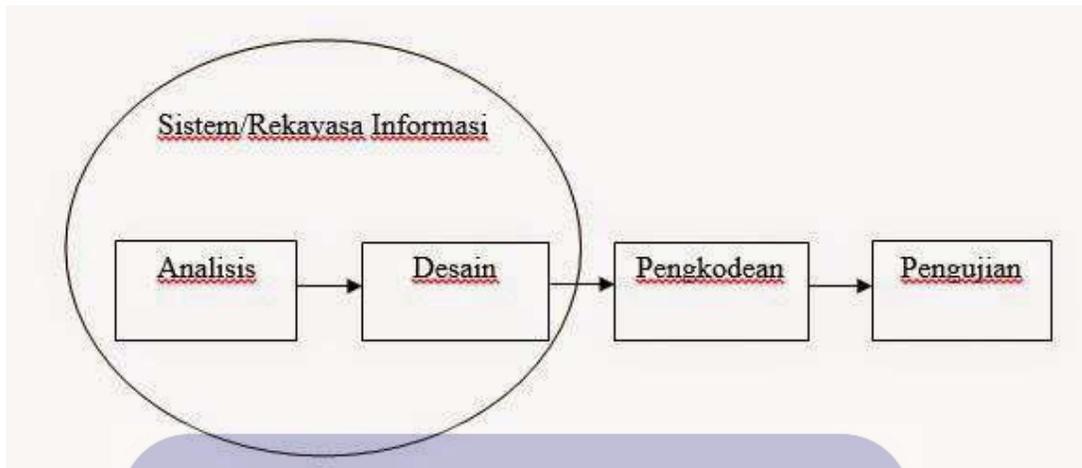
#### **2.1.6. Basis Data**

Menurut Sukanto dan M.Shalahuddin (2015:43) mengatakan bahwa, “Basis Data adalah sistem sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

#### **2.1.7. Model Pengembangan Perangkat Lunak**

Model sekuensial Linier atau sering disebut Model Pengembangan Air Terjun, merupakan paradigm model pengembangan perangkat lunak paling tua, dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian dan pemeliharaan.

Berikut adalah gambar model air terjun:



Sumber: Sukanto dan M.Shalahuddin (2015:29)

**Gambar II.1 Ilustrasi Metode Waterfall**

Dari berbagai kelebihan tahapan proses pengembangannya tetap pasti, mudah diaplikasikan, dan prosesnya teratur. Baik digunakan untuk software/program yang sudah jelas kebutuhan di awal, sehingga minim kesalahannya.

Dari berbagai kelemahannya proyek yang sebenarnya jarang mengikuti alur sekuensial seperti diusulkan, sehingga perubahan yang terjadi dapat menyebabkan hasil yang sudah didapatkan tim. pengembangan harus dirubah kembali/iterasi sering menyebabkan masalah baru.

## 2.2. Teori Pendukung

Teori pendukung digunakan untuk menggambarkan bentuk sistem secara terstruktur dan akurat dengan pendekatan analisis terstruktur. Analisa terstruktur

merupakan suatu metode analisa dengan menggunakan alat atau sarana (*tool*) yang mana sarana tersebut digunakan untuk membuat spesifikasi sistem terstruktur.

### 2.2.1. Diagram Alir Data (*Data Flow Diagram*)

#### 1. Konsep Dasar

Menurut Kendal (2010:265) “Suatu sistem secara keseluruhan dan beberapa subsistem bisa digambarkan secara grafis dengan kombinasi empat symbol”. Empat symbol yang dimaksud adalah :

#### 2. Simbol-Symbol yang digunakan

##### a. Kesatuan Luar (*Eksternal Entity*)

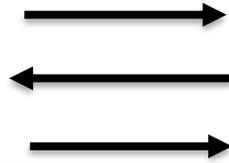
Dapat berupa bagian lain sebuah perusahaan, seseorang atau sebuah mesin yang dapat mengirim data atau menerima data dari sistem. Entitas eksternal, atau hanya entitas, disebut juga sumber atau tujuan data.

Sumber: Kendall (2010:265)

**Gambar II.2. Simbol Kesatuan Luar (*Eksternal Entity*)**

b. Arus Data (*Data Flow*)

Menunjukkan arus data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Arus data digambarkan dengan simbol tanda panah.

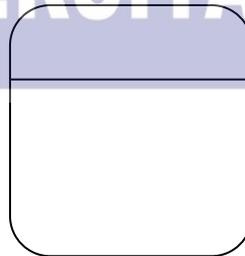


Sumber: Kendall (2010:265)

**Gambar II.3. Arus Data (*Data Flow*)**

c. Proses (*Process*)

Proses (*process*) menunjukkan adanya proses transformasi, dimana proses-proses tersebut selalu menunjukkan perubahan dalam didalam atau perubahan data jadi, aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk.

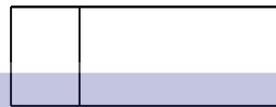


Sumber: Kendall (2010:265)

**Gambar II.4. Proses (*process*)**

d. Simpanan Data (*Data Store*)

Simpanan data menunjukkan tempat untuk penyimpanan data-data yang memungkinkan penambahan dan perolehan data, penyimpanan data mewakili seseorang, terdapat atau sesuatu, maka diberi nama dengan sebuah kata benda.



Sumber: Kendall (2010:265)

**Gambar II.5. Simpanan Data (*Data Store*)**

**3. Aturan Main**

- a. Dalam DAD tidak boleh menghubungkan antara *external entity* dengan *external entity* lainnya secara langsung.
- b. Dalam DAD tidak boleh menghubungkan antara *data store* dengan *data store* lainnya secara langsung.
- c. Dalam DAD tidak boleh menghubungkan *data store* dengan *external entity* secara langsung.
- d. Setiap proses harus ada *data flow* yang masuk dan juga *data flow* yang keluar.
- e. Aliran data tidak boleh terbelah menjadi dua atau lebih aliran data yang berbeda.

#### 4. Langkah-langkah mengembangkan DAD

- a. Membuat sebuah daftar tentang kegiatan-kegiatan bisnis dan digunakan untuk menentukan berbagai macam :
  - 1) Entitas eksternal
  - 2) Aliran data
  - 3) Proses-proses
  - 4) Penyimpanan data
- b. Menciptakan sebuah diagram yang menunjukkan entitas-entitas eksternal dan aliran-aliran data menuju sistem.
- c. Menggambarkan diagram nol yang menunjukkan proses-proses dan penyimpanan data.
- d. Menciptakan diagram anak untuk setiap proses dalam diagram nol.
- e. Mengecek kesalahan dan memastikan label-label yang ditetapkan untuk setiap proses dan aliran data.

#### 5. Tahapan Proses Pembuatan

Adapun langkah-langkah dalam proses pembuatan diagram alir data sebagai berikut:

- a. Diagram Konteks

Dalam Diagram Konteks adalah tingkat tertinggi dalam aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua eksternal yang ditunjukkan pada diagram konteks berikut aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat

penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data menuju dan dari sistem diketahui penganalisis dari wawancara dengan pengguna dan sebagai hasil analisis dokumen.

b. Diagram Nol

Diagram nol adalah pengembangan diagram konteks dan bisa mencakup sampai Sembilan proses. Memasukan lebih banyak proses pada level ini akan terjadi dalam suatu diagram yang kacau dan sulit dipahami. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data utama dari sistem (mewakili *file-file master*) dan semua entitas eksternal dimasukan kedalam diagram nol.

c. Diagram Detail

Merupakan diagram anak, tidak bisa menghasilkan keluaran atau menerima masukan dimana proses induknya juga tidak bisa menghasilkan atau menerima. Semua aliran data yang menuju atau keluaran dari proses induk harus ditunjukkan mengalir ke dalam atau keluar dari diagram anak.

### 2.2.2. Kamus Data (*Data Dictionary*)

#### 1. Konsep Dasar

Menurut Kendal (2006:333) Mendefinisikan, bahwa: Kamus data adalah suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Kamus data merupakan hasil refrensi data mengenai data (maksudnya, *metadata*) suatu data yang disusun oleh penganalisis sistem untuk membimbing

mereka selama melakukan analisis dan desain. Sebagai suatu dokumen, kamus data mengumpulkan dan mengkoordinasikan istilah-istilah data tertentu, dan menjelaskan apa arti setiap istilah yang ada.

Arus data yang ada didalam Diagram Alir Data sifatnya global dan hanya dapat mencerminkan keterangan yang jelas tentang data yang dicatat. Untuk maksud keperluan ini, maka kamus data harus memuat hal-hal berikut:

## 2. Hal yang harus dimuat dalam kamus data

### a. Nama Arus Data

Karena kamus data dibuat berdasarkan arus data yang mengalir di *data flow diagram*, maka nama dari arus data juga harus dicatat dikamus data, sehingga mereka yang membaca DFD dan memerlukan penjelasan yang lebih lanjut tentang suatu arus data tertentu di *data flow diagram* dapat langsung mencarinya dikamus data.

### b. Alias

Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang departemen yang satu dengan yang lainnya.

### c. Bentuk Data

Telah diketahui bahwa arus data dapat mengalir.

- 1) Dari kesatuan luar dari suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir.
- 2) Hasil dari suatu proses ke kesatuan luar, data yang mengalir ini biasanya terdapat di media laporan atau *query* tampilan layar atau dokumen hasil cetakan komputer.

- 3) Hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya berbentuk suatu *variable* atau parameter yang dibutuhkan oleh proses penerimanya.
- 4) Hasil suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu *variable*.
- 5) Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini laporan tercetak, tampil dilayar monitor, *variable*, parameter, field.

#### d. Arus Data

Arus data menunjukkan dari mana data mengalir akan dan kemana data akan menuju. Keterangan arus data ini perlu dikamus data supaya memudahkan mencari arus data dalam *flow diagram* (DFD)

#### e. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat dikamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

#### f. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat dikamus data karena dapat digunakan untuk mengidentifikasi kapan *input* data harus dimasukkan ke dalam sistem, kapan proses program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

#### g. Volume

Volume perlu dicatat didalam kamus data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya arus data

yang mengalir dalam suatu periode tertentu. Sedangkan volume puncak menunjukkan volume yang terbanyak.

#### h. Struktur Data

Struktur data menunjukkan arus data yang dicatat pada kamus data yang terdiri dari item-item atau elemen-elemen data.

### 3. Notasi Tipe Data

Untuk membuat spesifikasi format masukan dan keluaran suatu data.

**Tabel II.1.**  
**Notasi Tipe Data**

NOTASI	KETERANGAN
X	Setiap Karakter
9	Angka Numerik
A	Karakter Alphabet
Z	Angka Nol yang ditampilkan dalam spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	Hypen, sebagai tanda penghubung
/	Slash, sebagai tanda pembagi

Sumber: Sutabri (2012:172)

#### 4. Notasi Struktur Data

Untuk membuat spesifikasi elemen data.

**Tabel II.2.**  
**Notasi Struktur Data**

Notasi	Keterangan
=	Terdiri dari
+	And ( dan )
()	Pilihan ( ya atau tidak )
{ }	Iterasi / pengulangan proses
[ ]	Pilih salah satu pilihan
!	Pemisah pilihan di dalam tanda [ ]
*	Keterangan atau catatan
@	Petunjuk ( <i>key field</i> )

Sumber: Sutabri (2012:172)

#### 2.2.3. Entity Relationship Diagram (ERD)

Sukanto dan M. Shalahuddin (2014:289), “*Entity Relationship Diagram (ERD)* adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional”.

Sukanto dan M. Shalahuddin (2014:50), ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen). Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

### 1. Entitas atau *Entity*

Merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penanaman entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.



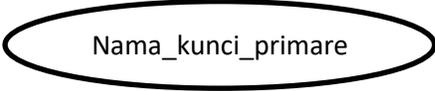
### 2. Atribut

*Field* atau kolom data yang butuh disimpan dalam suatu entitas.



### 3. Atribut Kunci Primer

Field atau kolom data yang butuh di simpan dalam entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id, kunci *primer* dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).



Nama\_kunci\_primare

Sumber : Sukamto (2014:50)

**Gambar II.8 Atribut Kunci Primer**

#### 4. Relasi

Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.



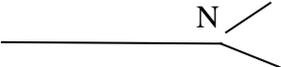
Nama\_relasi

Sumber : Sukamto (2014:50)

**Gambar II.9 Relasi**

#### 5. Asosiasi atau *Association*

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B.



N

Sumber : Sukamto (2014:50)

**Gambar II.10 Asosiasi atau *Association***

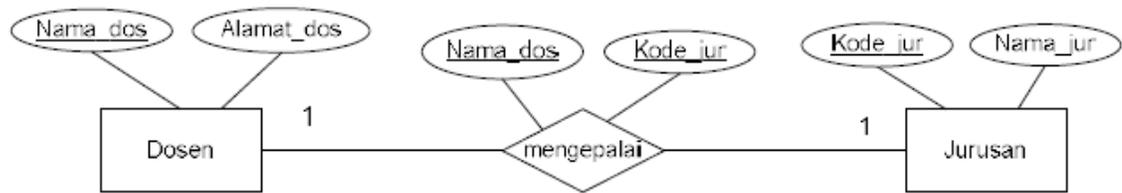
Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B maka ERD biasanya memiliki hubungan binary (satu relasi menghubungkan dua entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi ternary (satu relasi menghubungkan tiga buah relasi) atau N-ary (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan ternary atau N-ary.

#### A. Kardinalitas/Derajat Relasi

Berikut adalah contoh penggambaran relasi antar himpunan entitas lengkap dengan kardinalitas relasi dan atribut-atributnya:

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

Contoh, adanya relasi antara himpunan entitas Dosen dengan himpunan entitas Jurusan. Himpunan relasinya diberi nama “mengepalai”. Pada relasi ini, setiap Dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua Dosen yang menjadi ketua Jurusan). Dan setiap Jurusan pasti dikepalai oleh paling banyak satu orang Dosen. Maka penggambarannya adalah :

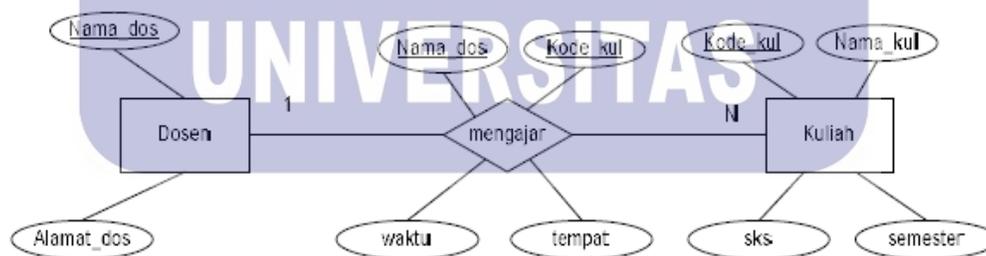


Sumber : Sukamto (2014:50)

**Gambar II.11 Relasi satu-ke-satu (*one-to-one*)**

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

Contoh, adanya relasi antara himpunan entitas Dosen dengan himpunan entitas Kuliah. Himpunan relasinya diberinama “mengajar”. Para relasi ini, setiap Dosen dapat mengajar lebih dari satu mata Kuliah, sedang setiap Dosen dapat mengajar lebih dari satu mata Kuliah, sedang setiap mata Kuliah diajar hanya oleh paling banyak satu orang Dosen. Maka penggambarannya adalah :

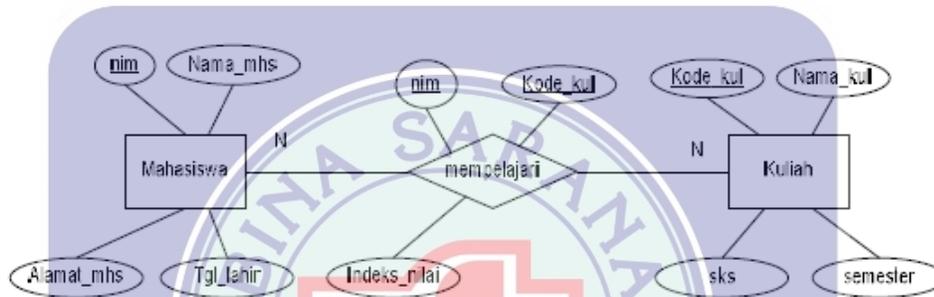


Sumber : Sukamto (2014:50)

**Gambar II.12 Relasi satu-ke-banyak (*one-to-many*)**

### 3. Relasi banyak-ke-banyak (*many-to-many*)

Contoh, adanya relasi antara himpunan entitas Mahasiswa dengan himpunan entitas Kuliah. Himpunan relasinya diberi nama “mempelajari”. Pada relasi ini, setiap Mahasiswa dapat mempelajari lebih dari satu mata Kuliah, demikian juga sebaliknya, setiap matakuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Maka penggambarannya adalah :



Sumber : Sukamto (2014:50)

**Gambar II.13 Relasi banyak-ke-banyak (*many-to-many*)**

#### 2.2.4. Logical Record Structure (LRS)

Menurut Kroenke (2006:76) mengemukakan bahwa “Logical Record Structure” adalah representasi dari struktur record-record pada tabel-tabel yang berbentuk dari hasil relasi antar himpunan entitas. Dibentuk dengan nomor dan tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Perbedaan LRS dan ERD adalah nama dan tipe *record* berada diluar *filed* tipe *record* di tempatkan. LRS terdiri dari link-link diantara tipe *record*. *Link* ini menunjukkan arah dari satu tipe *record* lainnya. Banyak link dari LRS yang diberi tanda *filed –filed*

yang kelihatan pada kedua link tipe *record*. Penggambaran LRS dimulai dengan menggunakan model yang dimengerti.

Dua metode yang digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode lain yang dimulai dengan ERD dan langsung dikonversikan ke LRS. Beda dengan diagram E-R nama tipe *record* berada diluar kotak *filed* tipe *record* ditempatkan. *Logical Record Structure* terdiri dari link-link diantara tipe *record*. Link ini menunjukkan arah dari satu tipe *record* lainnya. Banyak link dari LRS yang diberi tanda *filed -filed* yang kelihatan pada kedua link tipe *record*. Penggambaran LRS dimulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.



Sumber : Kroenke (2006:76)

**Gambar II.14 Logical Record Structure (LRS)**

### 2.2.5. Pengkodean

Menurut Mustakini (2014:384) Kode digunakan untuk memudahkan proses pengolahan data, karena dengan kode akan lebih mudah diidentifikasi. Kode dapat dibuat dengan suatu kerangka (*frame work*) yang menggunakan jangka angka atau huruf atau kombinasi angka dan huruf untuk memberi tanda terhadap kalsifikasi sebelumnya dibuat. Biasanya dalam proses akuntansi kode yang digunakan adalah angka, huruf, atau kombinasi keduanya.

Dalam pengkodean ada beberapa petunjuk pembuatan kode yang menjabarkan tentang petunjuk dari struktur kode yang baik, antara lain:

1. Harus Mudah Diingat

Agar kode dapat mudah diingat, maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan objek yang diwakili dengan kodenya.

2. Harus Unik

Kode harus unik untuk masing-masing item yang diwakilinya. Unik berarti tidak ada kode yang kembar.

3. Harus Fleksibel

Kode harus fleksibel sehingga memungkinkan perubahan-perubahan atau penambahan *item* baru dapat tetap diwakili oleh kode.

4. Harus Efisien

Kode harus sependek mungkin, selain mudah diingat juga akan efisien bila direkam, disimpan dalam komputer.

5. Harus Konsisten

Kode harus konsisten dengan kode yang telah dipergunakan

6. Harus Distandarisasi

Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi.

7. Spasi Dihindari

Spasi didalam kode sebaiknya dihindari, karena dapat menyebabkan kesalahan dalam menggunakannya.

8. Hindari Karakter yang mirip

Karakter-karakter yang hampir mirip serupa bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode

9. Panjang Kode Harus Sama

Masing-masing kode yang sejenis harus mempunyai panjang yang sama.

Tipe dari kode menurut Mustakini (2014:386) yaitu:

a. Kode Mnemonic (*mnemonic code*)

Kode mnemonic dibuat dengan dasar singkatan atau mengambil sebagian karakter dari *item* yang akan diwakili dengan kode ini.

b. kode urutan (*sequential Code*)

Kode yang nilai urutnya antara satu kode dengan kode berikutnya.

c. Kode Blok (*Block Kode*)

Kode ini mengklasifikasikan *item* ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode grup (*Group Code*)

Kode yang dibentuk berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

e. Kode Desimal (*Desimal Code*)

Kode ini mengklasifikasikan dalam bentuk 10 unit angka decimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai 99 tergantung banyaknya kelompok.

### 2.2.6. Pengertian HIPO

Menurut Puspitawati dan Sri Dewi Anggadini (2011:114) “Diagram HIPO (*Hierarchy Input Proses Output*)” serangkaian digram yang terdiri dari serangkaian level yang mengalir dari atas kebawah yang menggambarkan sistem yang lebih detail.”

HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program dan penggunaan HIPO ini mempunyai sasaran utama sebagai berikut:

- a. Untuk menyediakan suatu stuktur guna memahami fungsi dari sistem.
- b. untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukan *statement-statement* program yang digunakan untuk menjalankan fungsi tersebut.
- c. Untuk menyediakan penjelasan yang jelas dari *input* yang harus digunakan oleh *output* yang dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram.
- d. Untuk menyediakan *output* yang dapat dan sesuai dengan kebutuhan pemakai.