

## **BAB II**

### **LANDASAN TEORI**

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

##### **A. Sistem**

Menurut Tohari (2014:2) menjelaskan bahwa “Sistem adalah kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terkait, dan saling tergantung satu sama lain untuk mencapai tujuan”.

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, karakteristik tersebut menurut Tohari (2014:2) sebagai berikut :

##### **1. Komponen atau elemen (*Components*)**

Suatu sistem terdiri dari komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

##### **2. Batas Sistem (*Boundary*)**

Batasan sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Adanya batasan sistem, maka sistem dapat membentuk suatu kesatuan, karena dengan batas sistem ini, fungsi dan tugas dari subsistem satu dengan yang lainnya berbeda tetapi saling berinteraksi. Dengan kata lain, batas sistem merupakan ruang lingkup atau *scope* dari sistem atau subsistem itu sendiri.

### **3. Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar sistem adalah segala sesuatu diluar batas sistem yang mempengaruhi operasi suatu sistem. Lingkungan luar sistem dapat bersifat menguntungkan atau merugikan. Lingkungan luar sistem yang bersifat menguntungkan harus dipelihara dan dijaga supaya tidak hilang pengaruhnya. Sedangkan, lingkungan yang bersifat merugikan harus dihilangkan supaya tidak mengganggu operasi dari sistem.

### **4. Penghubung Sistem (*Interface*)**

Penghubung sistem merupakan suatu media (penghubung) antara satu subsistem dengan subsistem lainnya yang membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem lainnya. Dengan kata lain, melalui penghubung, *output* dari subsistem akan menjadi *input* bagi subsistem lainnya.

### **5. Masukan (*Input*)**

*Input* adalah energi atau sesuatu yang dimasukkan ke dalam suatu sistem yang dapat berupa masukan yaitu energi yang dimasukan supaya sistem dapat beroperasi atau masukan sinyal yang merupakan energi yang diproses untuk menghasilkan suatu luaran.

### **6. Luaran (*Output*)**

Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi luaran yang berguna, juga merupakan luaran atau tujuan akhir dari sistem.

### **7. Pengolah (*Process*)**

Suatu sistem mempunyai bagian pengolahan yang akan mengubah *input* menjadi *output*.

### **8. Sasaran (*Objective*)**

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan.

## **B. Sistem Informasi**

Sistem Informasi terdiri dari dua kata yaitu Sistem dan Informasi. Menurut Gondodiyoto (2007:109) Informasi “Merupakan data yang sudah diolah menjadi bentuk yang lebih berguna dan lebih berarti (bermanfaat) bagi penerimannya, menggambarkan suatu kejadian dan kesatuan nyata yang dapat dipahami dan dapat digunakan untuk pengambilan keputusan, sekarang maupun untuk masa depan”.

Dengan demikian menurut Gondodiyoto (2007:112) “Sistem Informasi dapat didefinisikan sebagai kumpulan elemen-elemen/sumberdaya dan jaringan prosedur yang saling berkaitan secara terpadu, terintegrasi dalam suatu hubungan hirarkis tertentu, dan bertujuan untuk mengolah data menjadi informasi”.

Komponen-Komponen Sistem Informasi yang disebut dengan istilah blok bangunan (*building block*) menurut Burch dan Gary Grudnitski dalam Gondodiyoto (2007:113) yaitu :

- a. Blok masukan (*input block*)
- b. Blok model (*model block*)

- c. Blok keluaran (*output block*)
- d. Blok teknologi (*technology block*)
- e. Blok basis data (*database block*)
- f. Blok pengendalian intern (*controls block*)

### C. Basis Data

Basis Data terdiri dari 2 kata yaitu Basis dan Data. Basis dapat diartikan sebagai tempat berkumpul, markas, atau gudang. Sedangkan Data merupakan kumpulan kejadian yang diangkat menjadi nyata atau dapat dikatakan data merupakan representasi fakta dari dunia nyata yang mewakili suatu objek.

Jika dijadikan kesatuan menurut Fathansyah (2012:2) Basis Data merupakan “Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”. Dan jika dihubungkan dengan sistem yang terkomputerisasi menurut Fathansyah (2012:3) Basis Data merupakan “Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik”.

Ada beberapa tipe basis data menurut Simarmata dan Iman Paryudi (2010:22) sebagai berikut :

1. Model basisdata file (*flat file database model*)
2. Model basisdata hierarki (*hierarchical database model*)
3. Model basisdata jaringan (*network database model*)
4. Model basisdata relasional (*relational database model*)
5. Model basisdata berorientasi objek (*object oriented database model*)

#### 6. Model basisdata relasi objek (*Object relational database model*)

Basis data yang berhubungan dengan komputer sangat erat hubungannya dengan Sistem Manajemen Basis Data (DBMS), menurut Ramakrishnan dan Gehrke (2003) dalam Simarmata dan Iman Paryudi (2010:8) DBMS “merupakan perangkat lunak yang di desain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar”.

Berikut Keuntungan DBMS adalah :

1. Mengurangi pengulangan data
2. Mencapai independensi data
3. Mengintegrasikan data beberapa file
4. Mengambil data dan informasi dengan cepat
5. Meningkatkan keamanan

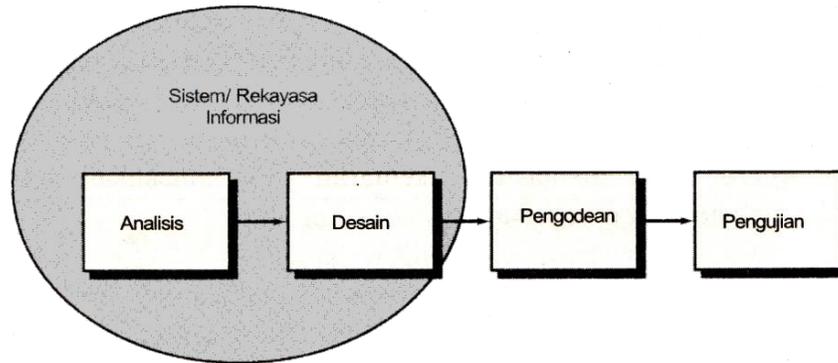
Sedangkan kerugian DBMS adalah :

1. Memperoleh perangkat lunak yang mahal
2. Memperoleh konfigurasi perangkat keras yang besar
3. Mempekerjakan dan mempertahankan staf DBA (para pengelola basisdata)

#### **D. Model Pengembangan Perangkat Lunak**

Model pengembangan perangkat lunak yang penulis terapkan dalam penulisan tugas akhir ini adalah Mode *Waterfall* (air terjun). Mode *Waterfall* merupakan bagian dari Model SDLC (*Software Development Life Cycle*) atau sering disebut juga (*System Development Life Cycle*) yang diartikan sebagai proses untuk mengembangkan dan mengubah perangkat lunak sebelumnya dalam suatu sistem

yang bertujuan untuk memperbaharunya sebagai pemenuhan kebutuhan dalam sistem dengan menggunakan model-model dan metodologi.



Sumber : A, S. Rossa dan Shalahuddin (2014:29)

**Gambar II.1 Ilustrasi Model *Waterfall***

Mode *Waterfall* itu sendiri menurut A. S, Rossa dan Shalahuddin (2013:28) adalah “Model SDLC air terjun (*waterfall*) sering disebut juga model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

Tahapan pendekatan mode *waterfall* (air terjun) menurut A. S, Rossa dan Shalahuddin (2013:29) sebagai berikut :

1. Analisis Kebutuhan

Proses pengumpulan kebutuhan yang dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat

lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu di dokumentasikan.

## 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

## 3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

## 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 5. Pendukung atau Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus

beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pembangunan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## **2.2. Teori Pendukung**

### **A. *Data Flow Diagram* (DFD)**

#### **1. Konsep Dasar**

Menurut Kendall dan Kendall (2014:263) “Diagram alir data menggambarkan pandangan sejauh mungkin mengenai masukan, proses, dan keluaran sistem, yang berhubungan dengan masukan, proses, dan keluaran dari model sistem umum”. Serta mempresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem yang lebih besar.

Menurut A. S, Rossa dan Shalahuddin (2013:69) “*Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured System Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson”.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya file kartu, *microfiche*, *hard disk*, *tape*, *diskette* dan lain sebagainya).

## 2. Simbol-Simbol yang digunakan

Dalam penggunaannya, DFD mempunyai simbol-simbol yang memiliki arti tertentu dalam menggambarkan bagian-bagian sistem, dan menurut Kendall dan Kendall (2014:265) simbol-simbol tersebut, yaitu :

### 1. Kesatuan Luar (*External Entity*)

Kotak rangkap dua digunakan untuk menggambarkan suatu entitas eksternal, misalnya sebuah perusahaan, seseorang atau sebuah mesin yang dapat mengirim data atau menerima data dari sistem.

### 2. Aliran Data

Tanda panah menunjukkan perpindahan data dari satu titik ke titik yang lain, dengan kepala tanda panah mengarah ke tujuan data.

### 3. Proses

Bujur sangkar dengan sudut membulat digunakan untuk menunjukkan adanya proses transformasi dan aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk.

### 4. Penyimpanan Data

Bujur sangkar dengan ujung terbuka, yang menunjukkan penyimpanan data. Penyimpanan data menandakan penyimpanan manual seperti lemari file atau sebuah file atau basisdata tekomputerisasi.

### 3. Aturan Pembuatan DFD

Ada beberapa aturan yang ditetapkan dan yang harus diperhatikan dalam pembuatan DFD, diantaranya :

1. *External Entity* tidak boleh dihubungkan secara langsung dengan *External Entity* yang lain.
2. *Data Store* tidak boleh dihubungkan secara langsung dengan *Data Store* yang lain.
3. *External Entity* tidak boleh dihubungkan secara langsung dengan *Data Store*.
4. Setiap proses harus ada *Data Flow* yang masuk dan *Data Flow* yang keluar.
5. Aliran data tidak boleh terbelah menjadi dua atau lebih aliran data yang berbeda.

### 4. Tahapan Proses Pembuatan

Dalam pembuatan DFD dibagi menjadi 3 diagram tahapan proses, yaitu :

#### 1. Diagram Konteks

Tingkatan tertinggi dalam diagram alir data. Dalam diagram konteks hanya memuat satu proses yang menunjukkan sistem secara keseluruhan (diberi nomor 0 pada proses), dalam diagram konteks tidak digambarkan penyimpanan data.

#### 2. Diagram Nol

Diagram nol menggambarkan detail dari diagram konteks. Pada diagram nol proses dibagi sesuai dengan prosedur sistem dengan ketentuan alir data masukan dan keluaran tetap konstan sesuai yang ada pada di diagram konteks.

Pada diagram nol sudah menunjukkan bentuk penyimpanan.

### 3. Diagram Detail

Diagram detail digambarkan untuk membuat atau menciptakan diagram anak yang lebih detail yang terdapat dalam proses pada diagram nol. Proses pada diagram nol yang dikembangkan disebut *parent process* (proses induk) dan diagram yang dihasilkan disebut *child diagram* (diagram anak). Apabila proses tersebut sudah tidak dapat lagi dikembangkan disebut proses *primitive*.

## B. Kamus Data

Menurut Mustakini (2014:725) “Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.”.

Pada tahap analisis, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari suatu arus data di DFD secara lebih terperinci dapat dilihat di kamus data.

Isi dalam kamus data menurut Mustakini (2014:726) sebagai berikut :

#### 1. Nama Arus Data

Karena kamus data dibuat berdasarkan arus data yang mengalir di DFD, maka nama dari arus data juga harus dicatat di kamus data, sehingga mereka yang

membaca DFD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DFD dapat langsung mencarinya dengan mudah di kamus data.

## **2. Alias**

Alias atau nama lain dari data dapat dituliskan bila nama lain ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

## **3. Bentuk Data**

Bentuk dari data perlu dicatat di kamus data, karena dapat digunakan untuk mengelompokkan kamus data ke dalam kegunaannya sewaktu perancangan sistem. Kamus data yang mencatat data yang mengalir dalam bentuk dokumen dasar atau formulir akan digunakan untuk merancang bentuk *input* sistem. Kamus data yang mencatat data yang mengalir dalam bentuk laporan tercetak dan dokumen hasil cetakan komputer akan digunakan untuk merancang *output* yang akan dihasilkan oleh sistem.

## **4. Arus Data**

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju. Keterangan data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini di DFD.

## **5. Penjelasan**

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

## 6. Periode

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

## 7. Volume

Volume yang perlu dicatat di kamus data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak.

## 8. Struktur Data

Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

Dalam kamus data terdapat notasi struktur data yang digunakan untuk membuat spesifikasi element data. Notasi yang umum digunakan antara lain :

**Tabel II.1**

### Notasi Struktur Data dalam Kamus Data

Notasi	Arti
=	Terbentuk dari atau terdiri dari atau sama dengan
+	<i>And</i> (dan)
[ ]	Salah satu dari (memilih salah satu dari elemen-elemen data di dalam kurung bracket ini)
	Sama dengan symbol [ ]

M{ }M	Iterasi (elemen data di dalam kurung brace beriterasi mulai minimum N kali dan maksimum M kali)
()	Optional
*	Keterangan setelah tanda ini adalah komentar

Sumber : Mustakini (2015:730)

### C. *Entity Relationship Diagram (ERD)*

#### 1. Konsep Dasar

*Entity Relationship Diagram* merupakan sebuah gambar atau bentuk diagram yang mempresentasikan mengenai hubungan relasi antar entitas yang disertai dengan attribute-atributnya dalam sebuah sistem.

Fathansyah (2012:81) menerangkan mengenai ERD bahwa : Model *Entity-Relationship* yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari 'dunia nyata' yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R).

#### 2. Notasi Simbolik

Notasi-notasi simbolik di dalam Diagram E-R yang digunakan adalah:

- a. Persegi panjang, menyatakan Himpunan Entitas.
- b. Lingkaran atau Elip, menyatakan Atribut (Atribut yang berfungsi sebagai *key* digaris bawah).
- c. Belah ketupat, menyatakan Himpunan Relasi.
- d. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.

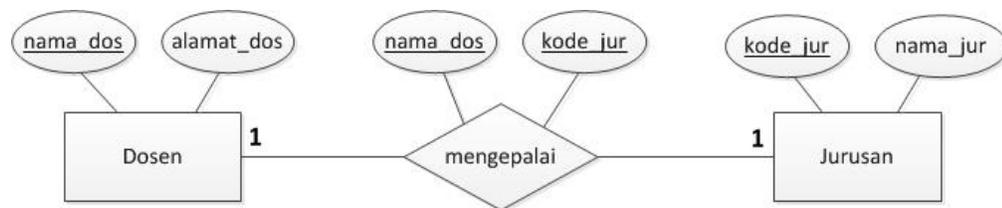
- e. Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).

### 3. Kardinalitas Relasi

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

Dapat diberi contoh sebagai berikut :

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. Penggambarannya sebagai berikut :



Sumber : Fathansyah (2012:82)

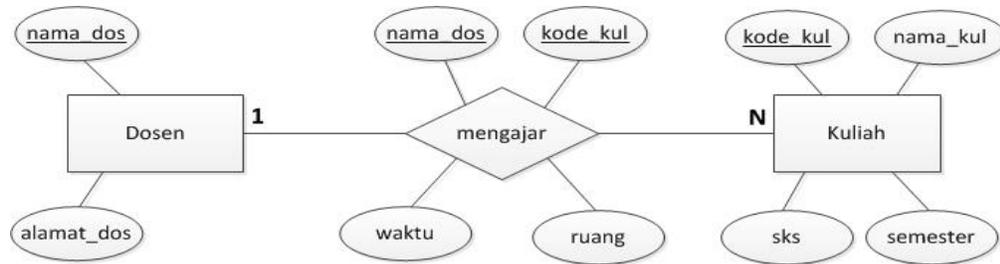
**Gambar II.2 Diagram E-R untuk Relasi Satu Ke Satu**

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

Dapat diberi contoh sebagai berikut :

Adanya relasi antara himpunan entitas Dosen dengan himpunan entitas Matakuliah. Himpunan relasinya kita beri nama “Mengajar”. Pada relasi ini, setiap dosen dapat mengajar lebih dari satu mata kuliah, sedangkan setiap

matakuliah diajar hanya oleh paling banyak 1 orang dosen. Penggambarannya sebagai berikut :



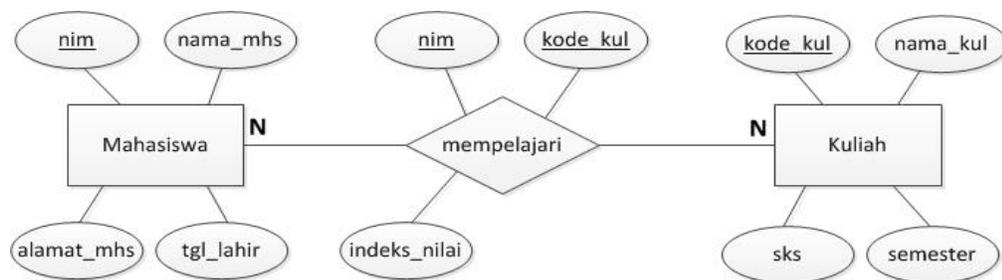
Sumber : Fathansyah (2012:83)

**Gambar II.3 Diagram E-R untuk Relasi Satu Ke Banyak**

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

Dapat diberi contoh sebagai berikut :

Adanya relasi antara himpunan entitas Mahasiswa dengan himpunan entitas Matakuliah. Himpunan relasinya diberi nama “Mempelajari”. Pada relasi ini, setiap mahasiswa dapat mempelajari lebih dari satu matakuliah. Demikian juga sebaliknya, setiap matakuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Penggambarannya sebagai berikut :



Sumber : Fathansyah (2012:84)

**Gambar II.4 Diagram E-R untuk Relasi Banyak Ke Banyak**

#### **D. *Logical Record Structure (LRS)***

Menurut Simarmata (2007:107) “*Logical Record Structure* adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antara himpunan entitas.”. Menentukan kardinalitas, jumlah tabel dan *foreign key* kardinalitas suatu hubungan menyatakan sejumlah kejadian terkait untuk masing-masing dua entitas. Jenis dasar konektivitas untuk hubungan adalah :

1. *One to one* (1:1)

Terjadi jika sebanyak satu kejadian dari suatu entitas A dihubungkan dengan satu kejadian entitas B.

2. *One to many* (1-N)

Terjadi ketika dalam satu kejadian dari entitas A, terdapat nol, satu atau banyak kejadian pada entitas B. Namun untuk satu kejadian dari entitas B, hanya ada satu kejadian pada entitas A. Contohnya suatu departemen memiliki banyak karyawan. Masing-masing karyawan ditugaskan dalam satu departemen.

3. *Many to many* (M-N)

Terkadang disebut tidak spesifik ketika untuk suatu kejadian dari entitas A, ada nol, satu, atau banyak kejadian dari entitas B dan untuk satu kejadian dari entitas B ada nol, satu, atau banyak kejadian dari entitas A. sebagai contoh karyawan ditugaskan dalam tidak lebih dari dua proyek pada waktu yang sama.

## E. HIPO

Mustakini (2014:787) menerangkan mengenai HIPO sebagai berikut :  
HIPO (*Hierarchy plus Input-Process-Output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program. Akan tetapi sekarang, HIPO juga banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu tiap-tiap modul di dalam sistem digambarkan oleh fungsi utamanya.

Ada tiga tingkatan HIPO berdasarkan fungsinya, menurut Mustakini (2014:788) adalah :

### 1. *Visual Table of Contents (VTOC)*

Diagram ini menggambarkan hubungan dari fungsi-fungsi di sistem secara berjenjang.

### 2. *Overview Diagrams*

*Overview Diagrams* menunjukkan secara garis besar hubungan dari *input*, proses dan *output*. Bagian *input* menunjukkan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

### 3. *Detail Diagrams*

*Detail Diagrams* merupakan diagram tingkatan yang paling rendah di diagram HIPO. Diagram ini berisi dengan elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

## F. Pengkodean

Pengkodean merupakan pemberian kode atau identitas *unique* (unik) pada suatu data agar data dapat dikenali dengan mudah dan dapat dibedakan dengan data-data yang lainnya. Fungsi kode menurut Mustakini (2014:384) “Kode digunakan untuk mengklasifikasikan data, memasukan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya”.

Ada beberapa hal yang harus diperhatikan dalam pembuatan kode, menurut Mustakini (2014:384) adalah :

1. Harus mudah diingat.
2. Harus unik.
3. Harus fleksibel.
4. Harus efisien.
5. Harus konsisten.
6. Harus distandarisasi.
7. Spasi dihindari.
8. Hindari karakter yang mirip.
9. Panjang kode harus sama.

Kode dibagi menjadi beberapa tipe kode, menurut Mustakini (2014:386) sebagai berikut :

### 1. Kode Mnemonik

Kode Mnemonik (*mnemonic code*) digunakan untuk tujuan supaya mudah diingat. Kode mnemonik dibuat dengan dasar singkatan atau mengambil

sebagian karakter dari item yang akan diwakili dengan kode ini. Misalnya kode “P” untuk mewakili Pria dan kode “W” untuk mewakili Wanita.

## **2. Kode Urut**

Kode urut (*sequential code*) disebut juga dengan kode seri (*serial code*) merupakan kode yang nilainya urut antara satu dengan kode berikutnya.

## **3. Kode Blok**

Kode blok (*block code*) mengklasifikasikan item ke dalam kelompok blok tertentu yang mencerminkan suatu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

## **4. Kode Group**

Kode group (*group code*) merupakan kode yang berdasarkan field-field dan tiap-tiap field kode mempunyai arti.

## **5. Kode Desimal**

Kode desimal (*decimal code*) mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.