

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Wahyudi (2012:3) menjelaskan, *computer* adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian di proses (*process*) untuk menghasilkan data dalam bentuk lain (*output*). Data adalah suatu file ataupun field yang berupa karakter atau tulisan dan gambar.

Pratama (2014:16) menjelaskan, *Computer* merupakan mesin elektronik untuk perhitungan cepat, yang mampu menerima inputan data digital, melakukan pemrosesan di memori sesuai dengan aplikasi yang digunakan, untuk kemudian dihasilkan menjadi sebuah informasi.

Pratama (2014:16) menjelaskan, *Computer* merupakan suatu alat elektronik yang mampu melakukan keempat tugas berikut: menerima inputan, memproses inputan, menyimpan perintah dan hasil pemrosesan, serta menyediakan *output* berupa informasi.

##### 2.1.2. Pengertian Data

Sutabri (2012:2) menjelaskan, Data adalah bahan mentah yang diproses untuk menyajikan informasi.

**Tabel 2.1 Tipe-tipe data**

No	Tipe	Keterangan
1.	Text	sebagian besar karakter yang ada pada keyboard, termasuk angka; biasanya dibatasi hingga 255 karakter
2.	Memo	blok-blok besar berisi teks
3.	Number(atau numeric)	hanya bilangan, tanpa karakter teks

### Lanjutan Tabel 2.1 Tipe-tipe data

No	Tipe	Keterangan
4.	Currency	pada dasarnya bilangan numeric dengan empat tempat decimal dan sebuah lambang mata uang
5.	AutoNumber	nomor yang secara otomatis ditambahkan untuk setiap record baru
6.	Yes/No	untuk informasi terpisah yang bisa dibagi ke dalam dua kategori; seperti True/False, Yes/No, Up/Down
7.	OLE Object	suatu objek seperti lembar-kerja Microsoft Excel, dokumen Microsoft Word, atau informasi grafis
8.	Hyperlink	untuk menyimpan URL (Uniform Resource Locator)

#### 2.1.3. Pengertian Informasi

Pratama (2014:393) menjelaskan, Informasi didefinisikan sebagai hasil dari pengolahan data memanfaatkan teknologi yang tersedia, untuk menghasilkan pengetahuan serta memberikan arti dan nilai.

Pratama (2014:393) menjelaskan, Informasi didefinisikan sebagai hasil dari pengolahan, manipulasi, dan pengorganisasian data, sehingga menambah nilai dan pengetahuan di dalamnya.

Pratama (2014:393) menjelaskan, Informasi merupakan jawaban atas berbagai pertanyaan dan persoalan serta pengetahuan yang diinformasikan dan diterima mengenai fakta dan keadaan tertentu.

#### 2.1.4. Metode Pengembangan Sistem

Sukanto dan Shalahuddin (2013:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut 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 mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

a. Analisis

Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh *user*.

b. Desain

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

c. Pengodean

Pada tahap pengodean, desain harus ditranlasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain

d. Pengujian

Tahap pengujian fokus pada sistem dari segi logika 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.

e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah sistem mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau sistem harus beradaptasi dengan lingkungan baru. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang sudah ada, tapi tidak untuk sistem baru.

## 2.2. Teori Judul

### 2.2.1. Pengertian Aplikasi

Asropudin (2013:6), aplikasi (*application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu.

Sutabri (2012:148), aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

### **2.2.2. Pengertian Web**

Simarmata (2010:31) Aplikasi Web adalah sistem perangkat lunak yang berdasarkan teknologi dan *world wide web* mereka menyediakan sumber daya web spesifik, seperti konten dan layanan melalui antar muka antara pengguna dan *browser web*.

### **2.2.3. Pengertian Helpdesk**

Menurut Santoso (2014:2) *Helpdesk* sebagai Single Point Of Contact (SPOC) menjadi fasilitas komunikasi antara pelanggan atau pengguna dengan tim pendukung di perusahaan penyedia produk atau jasa.

Berdasarkan definisi diatas *helpdesk* merupakan bagian struktur atau program yang menangani pertanyaan dan keluhan baik itu dari pihak internal atau pihak eksternal dengan menyediakan informasi atau solusi yang dibutuhkan pengguna.

### **Kelebihan Helpdesk**

*Helpdesk* memberikan pelayanan terbaik pada penggunanya serta dapat menurangi biaya. Hal ini dikarenakan berbagai keuntungan penggunaan *helpdesk*.

1. *Helpdesk* dapat memberikan solusi atas pertanyaan – pertanyaan dalam kurun waktu singkat.
2. *Helpdesk* dapat mengecek permasalahan yang ada dan mengatur pembagian staff.
3. *Helpdesk* dapat meningkatkan efisiensi perusahaan dalam menangani pertanyaan dan keluhan pelanggan.
4. *Helpdesk* dapat memberi laporan kerja perkembangan kinerja para staff dan pimpinan.
5. *Helpdesk* dapat menangani pertanyaan dan keluhan yang sejenis karena pertanyaan dan keluhan dicatat.

#### 2.2.4. Pengertian PDAM Tirta Musi Palembang

Perusahaan Daerah Air Minum (PDAM) Tirta Musi Palembang merupakan Badan Usaha Milik Daerah (BUMD) yang didirikan pada tanggal 3 April 1976. Perusahaan ini bertujuan memberikan pelayanan penyediaan air minum kepada masyarakat kota Palembang dengan kualitas dan kuantitas yang sesuai dengan standar yang ditetapkan.

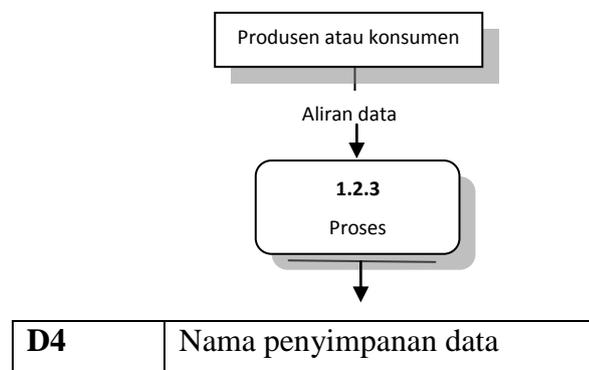
#### 2.2.5. Aplikasi *Helpdesk* Berbasis *Web* Pada PDAM Tirta Musi Palembang

Aplikasi *Helpdesk* sebagai Alat bantu untuk memudahkan operator tiap unit untuk mengadakan kerusakan alat-alat computer *software* dan *hardware* yang ada serta memudahkan admin untuk memproses pengaduan tersebut berbasis web.

### 2.3. Teori Khusus

#### 2.3.1. Pengertian DFD (*Data Flow Diagram*)

Sukamto dan Shalahuddin (2014:69) menjelaskan, “*Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. System yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah system.” Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson:



**Gambar 2.1** Contoh DFD yang dikembangkan Chris Gane dan Trish Sarson

---

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis system perangkat lunak untuk system perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

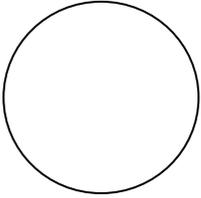
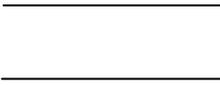
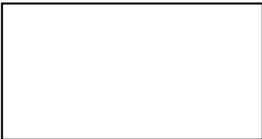
Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk merepresentasikan sebuah system atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD tidak sesuai untuk memodelkan system perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampuradukkan pemrograman terstruktur dan pemrograman berorientasi objek.

Rosa dan Shalahuddin (2014:71) menjelaskan, “notasi- notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

**Tabel 2.2 Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)**

No	Notasi	Keterangan
1		<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2		<p><i>File</i> atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM))</p>
3		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>

---

**Lanjutan Tabel 2.2 Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)**

No	Notasi	Keterangan
4		<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p>

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan system yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun system lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara system yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam system yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah system, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di- *breakdown*.

#### 4. Membuat DFD Level 3 dan seterusnya

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

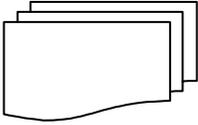
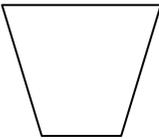
Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

#### 2.3.2. Pengertian Blockchart

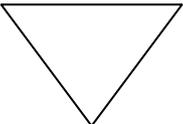
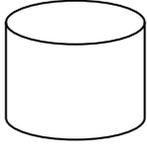
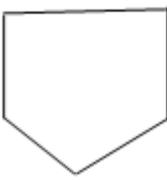
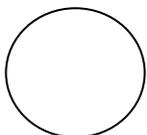
Kristanto (2008:75) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Kristanto (2008:75) menjelaskan, “Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

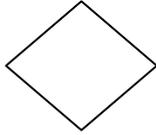
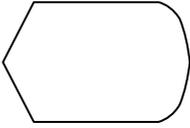
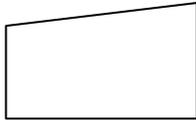
**Tabel 2.3 Simbol-simbol dalam *Blockchart***

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses Manual

**Lanjutan Tabel 2.3 Simbol-simbol dalam *Block Chart***

No	Simbol	Keterangan
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan ( <i>Storage</i> )
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.

**Lanjutan Tabel 2.3 Simbol-simbol dalam *Block Chart***

No	Simbol	Keterangan
11.		Pengambilan keputusan ( <i>Decision</i> ).
12.		Layar peraga ( <i>monitor</i> ).
13.		Pemasukkan data secara manual.

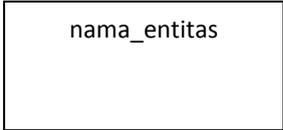
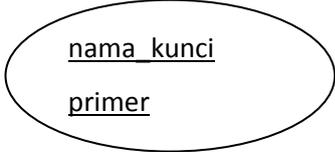
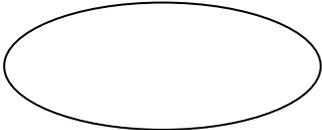
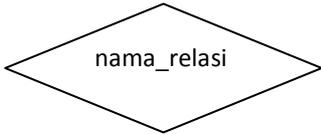
### 2.3.3. Pengertian ERD (*Entity Relationship Diagram*)

Rosa dan Shalahuddin (2014 : 50) menjelaskan, tentang pemodelan awal basis data yang paling banyak digunakan adalah:

menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. 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.

Shalahuddin, *et al* (2014 : 71) menjelaskan, "Berikut adalah symbol-simbol yang digunakan pada ERD dengan notasi Chen:

**Tabel 2.4 Simbol-Simbol ERD Dengan Notasi Chen**

No	Simbol	Deskripsi
1	Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> 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)
4	Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
5	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja

**Lanjutan Tabel 2.4 Symbol-simbol ERD dengan notasi Chen**

No	Simbol	Deskripsi
6	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas

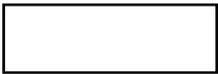
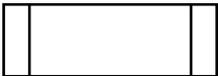
ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

#### 2.3.4. Pengertian *Flowchart*

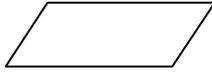
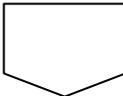
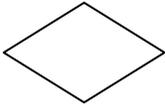
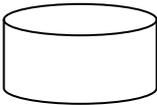
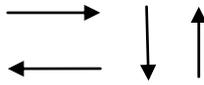
Indrajani (2015:36), "*Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.

Indrajani (2015:36) menjelaskan tentang simbol-simbol yang sering digunakan dalam *flowchart* dapat dilihat pada tabel berikut ini:

**Tabel 2.5 Simbol-Simbol Dalam *Flow Chart***

No.	Simbol	Keterangan
1.		Simbol <b>Start</b> atau <b>End</b> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.
3.		Simbol yang menyatakan bagian dari program (sub program).
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.

Lanjutan Tabel 2.5 Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
5.		Simbol <b>Input/Output</b> yang mendefinisikan masukan dan keluaran proses.
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>desicion</i> (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).
15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

### 2.3.5. Pengertian Kamus Data

Rosa dan Shalahuddin (2014:73) menjelaskan, “Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada system perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

- a. Nama – nama dari data
- b. Digunakan pada – merupakan proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data
- d. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Kamus data memiliki beberapa symbol untuk menjelaskan informasi tambahan sebagai berikut:

**Tabel 2.6 Simbol-Symbol Dalam Kamus Data**

No	Simbol	Keterangan
1	=	disusun atau terdiri dari
2	+	Dan
3	[]	baik ...atau...
4	{ } <sup>n</sup>	n kali diulang/ bernilai banyak
5	( )	data opsional
6	*...*	batas komentar

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

## 2.4. Teori Program

### 2.4.1. Sekilas Tentang Basis Data (*Database*)

Rosa dan Shalahuddin (2014:43) menjelaskan, “System basis data adalah system 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.

Indrajani (2014:70), sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi.

### 2.4.2. Sekilas Tentang *HTML*

Pratama (2014:218) menjelaskan, “HTML atau *Hyper Text Markup Language* merupakan salah satu bahasa pemrograman yang digunakan di dalam pembuatan aplikasi dan layanan berbasis web.”

**Tabel 2.7 Tag-Tag Dasar HTML**

Tag Awal	NN	IE	W3	Kegunaan
<html>	3.0	3.0	3.2	Definisi sebuah dokumen HTML
<body>	3.0	3.0	3.2	Definisi body dokumen
<h1><h6>	3.0	3.0	3.2	Definisi heading 1 s.d. 6
<p>	3.0	3.0	3.2	Definisi sebuah paragraf
 	3.0	3.0	3.2	Sisipkan line break
<hr>	3.0	3.0	3.2	Definisi sebuah horizontal rule
<!-->	3.0	3.0	3.2	Definisi komentar dalam source code

### 2.4.3. Pengertian *MySQL*

Kadir (2013:15) menjelaskan, “MySQL adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan

penyimpanan dan pengaksesan data. Dengan menggunakan MySQL, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.”

Badiyanto (2013:57), “MySQL merupakan sebuah database server *SQLmultiuser* dan *multi threaded*.”

Madcoms (2011:140), *MySQL* adalah salah satu program yang dapat digunakan sebagai database, dan merupakan salah satu *software* untuk database server yang banyak digunakan di beberapa platform misalnya *windows, linux, dll*.

*MySQL* memiliki beberapa kelebihan, antara lain:

1. *MySQL* dapat digunakan oleh beberapa *user* dalam waktu bersamaan tanpa mengalami masalah.
2. *MySQL* memiliki kecepatan yang bagus dalam menangani *query* sederhana.
3. *MySQL* memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *where* dalam perintah *Query*.
4. *MySQL* memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perjanjian yang mendetail serta sandi terenkripsi.
5. *MySQL* mampu menangani basis data dalam skala besar dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta kurang lebih 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.
6. *MySQL* dapat melakukan koneksi dengan *client* menggunakan *protocol TCP/IP, Unix Scket (UNIX)* atau *Named Pipes (NT)*.
7. *MySQL* dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa.
8. *MySQL* dapat berjalan stabil dari berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.

*MySQL* didistribusikan secara *open source*, dibawah lisensi *GPL* sehingga dapat digunakan secara gratis.

#### **2.4.4. Sekilas Tentang PHP**

Nugroho (2013:153) menjelaskan, “PHP kepanjangan dari *Hypertext Preprocessor*) itu bahasa pemrogramana berbasis web. PHP itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis web (website, blog, atau aplikasi web).”

Abdul (2013:120) menjelaskan, “PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi web. Ditinjau dari pemrosesannya, PHP tergolong berbasis *server side*. Artinya, pemrosesan dilakukan di *server*.”

#### **2.4.5. Sekilas tentang Dreamweaver CS6**

Sadeli (2014:12), “Dreamweaver merupakan suatu perangkat lunak web editor keluaran adobe system yang digunakan untuk mengembangkan dan mendesain suatu website dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya.

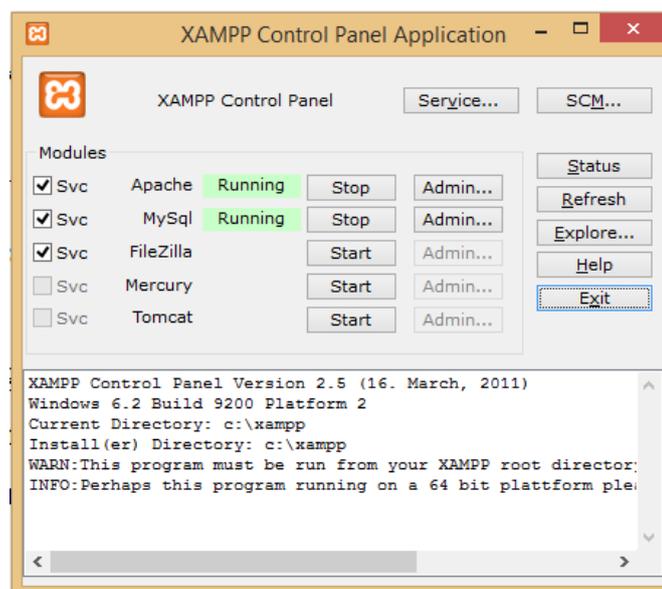
#### **2.4.6. Pengenalan XAMPP**

Nugroho (2013 : 1) menjelaskan, XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legas.”

Dibawah folder utama xampp, terdapat beberapa folder penting yang perlu diketahui. Penjelasan fungsinya sebagai berikut:

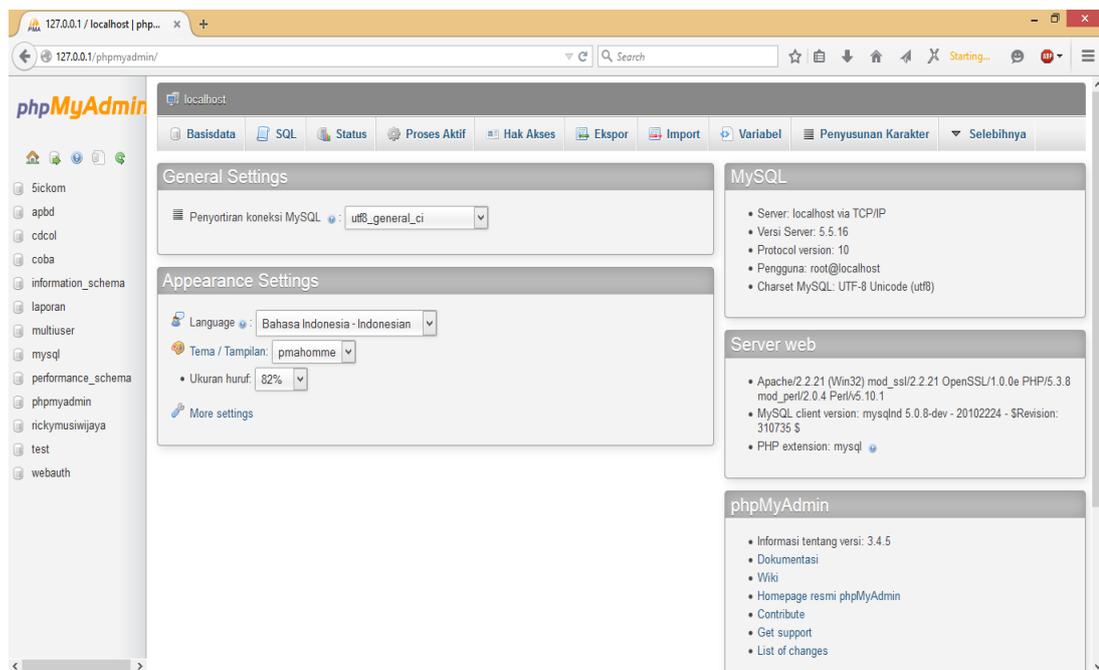
**Tabel 2.8 Folder-Folder Penting Dalam XAMPP**

No	Nama Folder	Fungsi
1.	Apache	Folder utama dari <i>Apache Webserver</i> .
2.	Htdocs	Folder utama untuk menyimpan data-data latihan <i>web</i> , baik PHP maupun HTML biasa. Pada folder ini, anda dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua folder dan file program di htdocs bisa diakses dengan mengetikkan alamat <a href="http://localhost/">http://localhost/</a> di browser.
3.	Manual	Berisi subfolder yang di dalam terdapat manual program dan <i>database</i> , termasuk manual PHP dan MYSQL.
4.	Mysql	Folder utama untuk <i>database</i> MYSQL server. di dalamnya terdapat subfolder data (lengkapnya: C:\xampp\mysql\data) untuk merekam semua nama <i>database</i> , serta subfolder bin yang berisi <i>tools</i> klien dan server MYSQL.
5.	Php	Folder utama untuk program PHP.


**Gambar 2.2 Tampilan XAMPP**

### 2.4.7. Pengertian phpMyAdmin

Nugroho (2013:71) menjelaskan, “phpMyAdmin adalah tools *MySQL Client* berlisensi *Freeware*, Anda dapat mengunduhnya secara gratis pada situs resminya di [www.phpmyadmin.net](http://www.phpmyadmin.net).



**Gambar 2.3** Tampilan phpMyAdmin

### 2.4.8. Sekilas Tentang CSS

Salah dkk (2007:53) menjelaskan tentang CSS (*Cassading Style Sheet*) adalah:

Template yang mengontrol pemformatan tag HTML pada halaman web anda. Bila anda telah terbiasa menggunakan Microsoft Word, anda dapat melihat bahwa konsep Style Sheet mirip dengan template pada Microsoft Word: anda dapat mengubah penampilan pada dokumen Word dengan mengubah format pada style dokumen. Begitu pula anda dapat mengubah penampilan halaman web dengan mengubah format pada tag HTML tertentu melalui Style Sheet, untuk selanjutnya menggantikan spesifikasi default dari browser untuk tag-tag tersebut.

Betha dan Husni (2014 : 132) menjelaskan, “Secara umum disusun oleh tiga bagian yaitu, selector (elemen yang akan didefinisikan), property (atribut yang akan diubah) dan nilai sebagaimana berikut:

```
Selector {property : value}
```

Antara property dan nilai disahkan dengan titik-dua (colon) seperti contoh di bawah ini:

```
Body {color : black}
```

Jika ilai berupa beberapa kata, gunakan tanda petik ganda:

```
p{font - family : “sans serif”}
```

Jika lebih dari satu property pisahkan dengan titik-koma (semi colon):

```
p{text – align : center; color : red}
```

Jika ingin lebih mudah dibaca sebagaimana berikut:

```
p
{
text – align : center ;
color : black ;
font – family : arial
}
```

Jika selector dikelompokkan:

```
H1, H2, H3, H4, H5, H6
{
color : green ;
}
```

Jika menggunakan atribut class (dalam definisi CSS):

```
P.Kanan {text – align : right}
P.Kiri {text – align : center}
```

Berikut ini sintaks yang ditulis dalam dokumen html menggunakan definisi di atas:

```
<P class=”kanan”> This Paragraph will be right aligned <p>
```

<P class="kiri"> This Paragraph will be center aligned <p>

Jika menggunakan ID atribut:

```
#Kanan {text – align : right}
```

#### 2.4.9. Sekilas Tentang JavaScript

Betha dan Husni (2014:267) menjelaskan, “JavaScript merupakan modifikasi dari bahasa c++ dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan asp ataupun internet explorer. Secara khusus beberapa hal yang penting dalam JavaScript adalah:

- Menggunakan blok awal “{“ dan blok akhir “}”
- Automatic conversion dalam pengoperasian tipe data yang berbeda
- Sensitive case, sehingga programmer java harus ekstra hati-hati dalam menggunakan nama variable, fungsi dan lain-lain
- Extension umumnya menggunakan “\*.js”
- Setiap statement dapat diakhiri dengan “;” sebagaimana halnya c++ tetapi dapat juga tidak
- Jika tidak didukung oleh browser versi lama, scriptnya dapat disembunyikan di antara tag “<!--” dan “-->”
- Jika program dalam satu baris terlalu panjang dapat disambung ke baris berikut dengan karakter “\”

Betha dan Husni (2014 : 272) menjelaskan, “Tipe data pada JavaScript umumnya sama pada setiap mesin, hal ini diakibatkan desain awal java memang dikhususkan untuk bisa berjalan pada semua mesin

**Tabel 2.9 Tipe data JavaScript**

Primitive Type	Size	Minimum	Maximum	Version Type
Boolean	1 bit	-	-	
Char	16 bit	Unicode 0	Unicode 2E16-1	
Byte	8 bit	-128	+127	Versi 1.1 keatas

**Lanjutan Tabel 2.9 Tipe data JavaScript**

<b>Primitive Type</b>	<b>Size</b>	<b>Minimum</b>	<b>Maximum</b>	<b>Version Type</b>
Short	16 bit	-2E-15	+2E15-1	Versi 1.1 keatas
Int	32 bit	-2E-31	+2E31-1	
Long	64 bit	-2E-63	+2E63-1	
Float	32 bit	IEEE754	IEEE754	
Double	64 bit	IEEE754	IEEE754	
Void	-	-	-	Versi 1.1 keatas