



BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Umum

2.1.1 Pengertian Komputer

Wahyudi (2012:3) mengemukakan, “komputer adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian di proses (*process*) untuk menghasilkan data dalam bentuk lain (*output*)”.

Asropudin (2013:19) menyatakan bahwa “komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan intruksi atau program yang tersimpan dalam memori masing-masing komputer”.

Jadi, komputer adalah suatu alat elektronik yang membantu melakukan beberapa tugas dan pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.

2.1.2 Pengertian Perangkat Lunak

Sukanto dan Shalahuddin (2014:2), “perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

2.1.3 Pengertian Data

Bahra (2013:20) mengatakan, “data merupakan komponen dasar yang akan diproses lebih lanjut untuk menghasilkan informasi”.

Dhanta (2009:212) menyatakan bahwa “data berasal dari kata *datum* (bahasa latin) yang berarti fakta. Kata data bersifat *plural*, sebagaimana kata air, udara dan sebagainya. Oleh karena itu kata data akan salah bila ditulis dengan data - data, banyak data dan sebagainya”.

Menurut Kristanto (2008:7) , “data adalah penggambaran dari sesuatu dan kejadian yang kita hadapi. Data juga berarti kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata”.



Jadi, data adalah kumpulan dari angka maupun karakter yang berasal dari kenyataan yang belum berarti dan harus diolah terlebih dahulu untuk menjadi suatu informasi yang lebih berarti bagi penggunanya.

2.1.4 Pengertian Pengolahan Data

Kristanto (2011:8) menjelaskan bahwa “pengolahan Data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data informasi yang memiliki kegunaan”.

Bahra (2013:9) menjelaskan bahwa “pengolahan Data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Jadi, pengolahan data merupakan proses pengolahan data yang *diinputkan* untuk diubah menjadi informasi yang berguna.

2.1.5 Pengertian Sistem

Kristanto (2008:1) mengatakan bahwa “sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

2.1.6 Karakteristik Sistem

Menurut Bahra (2013:3) sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, masukan, keluaran, pengolahan dan sasaran atau tujuan. Adapun karakteristik sistem yang dimaksud yaitu:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling berkerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem



memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus dijaga dan diperlihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media yang menghubungkan antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini kemungkinan sumber-sumber daya mengalir dari satu subsistem yang lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal *maintenance input*. Sinyal *maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal *input* adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

6. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan atau sistem itu sendiri sebagai pengolahannya. Pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, jika sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila



mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.1.7 Metode Pengembangan Sistem

Sukanto dan Shalahudin, (2014:28), menjelaskan tentang “Metode pengembangan sistem yaitu metode air terjun. 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 terurut mulai dari analisis, desain, pengkodean, pengujian, dan pemeliharaan”.

1. Analisis

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

2. Desain

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

3. Pengkodean

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

4. Pengujian

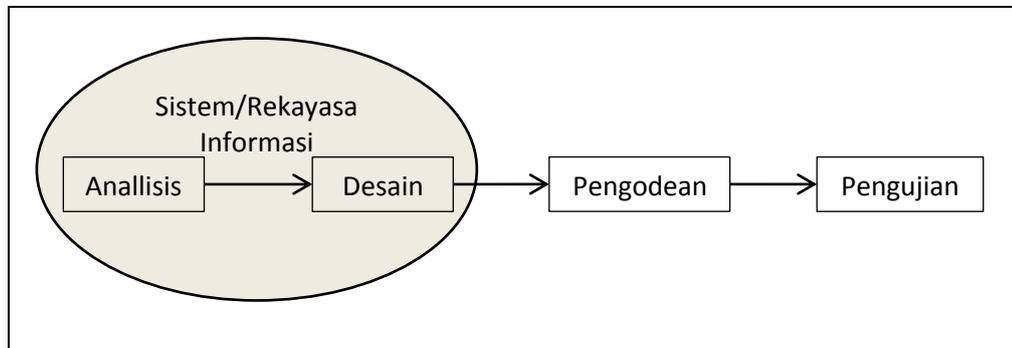
Tahap pengujian fokus pada sistem dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pemeliharaan

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.



Gambar 2.1 Ilustrasi Model *Waterfall*

2.1.8 Pengertian Aplikasi

Asropudin (2013:6) mengemukakan bahwa “*Application* adalah *Software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms. Word*, *Ms. Excel*. *Application* program adalah program komputer yang dibuat untuk mengerjakan atau menyelesaikan masalah-masalah khusus, misalnya program penggajian”.

Menurut Anindito, Kusnadi, dan Purnomo (2008:18) menyatakan bahwa “Aplikasi atau pemrograman aplikasi adalah pengguna sistem komputer yang membuat aplikasi-aplikasi pengguna yang dijalankan diatas sistem operasi”.

2.1.9 Pengertian Laporan

Nuraida (2008:69) mengemukakan bahwa “laporan adalah alat komunikasi tertulis yang memuat hasil pengolahan data dan informasi serta memberikan kesimpulan atau rekomendasi atas fakta-fakta atau keadaan-keadaan yang telah diselidiki sebelumnya”.

Menurut kamus Bahasa Indonesia (2008:818) laporan adalah segala sesuatu yang dilaporkan.

Jadi, laporan adalah alat komunikasi tertulis untuk memberikan informasi atas segala sesuatu yang dilaporkan.

2.1.10 Pengertian Pemakaian

Pemakaian atau penggunaan adalah suatu proses dari hasil atau bahan yang telah kita gunakan untuk keperluan dengan maksud dan tujuan tertentu.



2.1.11 Pengertian Lembar

Lembar adalah helaian suatu barang yang dapat digunakan dalam suatu proses, baik itu helaian benang maupun helaian kertas yang dapat diolah untuk mendapatkan suatu hasil yang diinginkan.

2.1.12 Pengertian Permintaan

Alam (2007:75) menyatakan bahwa “Permintaan adalah keinginan yang disertai kemampuan untuk membeli barang dan jasa pada tingkat harga dan waktu tertentu”.

Indrastuty (2011:28) menyatakan bahwa “Permintaan adalah jumlah barang/jasa yang akan dibeli pada berbagai tingkat, harga, waktu dan tempat tertentu”.

Jadi, permintaan merupakan jumlah barang/jasa yang dibutuhkan sesuai keinginan yang disertai kemampuan untuk mendapatkannya sesuai pada tingkat harga, waktu dan tempat tertentu.

2.1.13 Pengertian Obat

Sumardjo (2009:26) menyatakan bahwa “Obat adalah suatu bahan kimia yang dapat memengaruhi organisme hidup dan digunakan untuk keperluan diagnosis, pencegahan dan pengobatan suatu penyakit”.

Menurut kamus Bahasa Indonesia (2008:1013) obat adalah bahan yang digunakan untuk mengurangi, menghilangkan, atau menyembuhkan sakit.

Jadi, obat adalah suatu bahan kimia yang digunakan untuk mengurangi, menghilangkan dan menyembuhkan sakit yang dialami oleh penderita sakit.

2.1.14 Pengertian Puskesmas

Makhfudli dan efendy (2009:274) menyatakan bahwa “Puskesmas adalah suatu kesatuan organisasi kesehatan fungsional yang merupakan pusat pengembangan masyarakat yang juga membina peran serta masyarakat disamping memberikan pelayanan secara menyeluruh dan terpadu kepada masyarakat di wilayah kerjanya dalam bentuk kegiatan pokok”.



2.1.15 Pengertian Aplikasi Laporan Pemakaian dan Lembar Permintaan Obat pada Puskesmas 11 Ilir Palembang

Aplikasi laporan pemakaian dan lembar permintaan obat pada Puskesmas 11 Ilir Palembang adalah suatu aplikasi yang digunakan untuk mempermudah dan mempercepat dalam memproses laporan pemakaian dan lembar permintaan obat sehingga informasi yang dihasilkan lebih akurat dan efisien, dengan menggunakan program aplikasi *PHP* dan *database MySQL* sebagai media penyimpanan data.

2.2 Teori Khusus

2.2.1 Data Flow Diagram (DFD)

Menurut Kristanto (2008:61), “*Data Flow Diagram (DFD)* adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi data yang tersimpan dan proses yang dikenakan pada data tersebut”.

Bahra (2013:64) mendefinisikan bahwa “*DFD* adalah model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil.

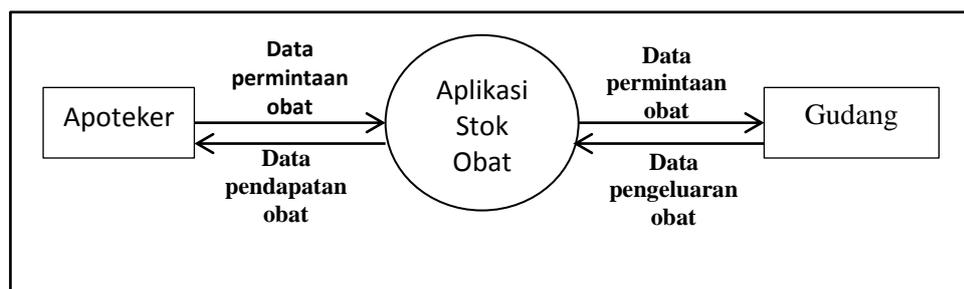
Dalam *DFD* ada 3 tingkatan tahap, yaitu:

1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari *DFD* yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

2. *DFD* Zero

DFD level 0 atau bisa juga disebut diagram konteks merupakan gambaran bagaimana sistem berinteraksi dengan external entity.



Gambar 2.2 *DFD* Level Zero

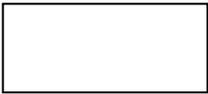
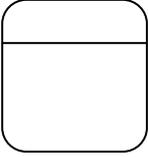
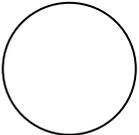
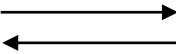
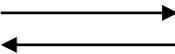
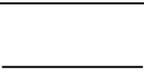


3. Daftar Kejadian

Daftar kejadian digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi di dalam lingkungan sehari-hari membutuhkan tanggapan atau respon dari sistem”.

Ada beberapa simbol *DFD* yang dipakai untuk menggambarkan data beserta proses transformasi data, antara lain:

Tabel 2.1 Simbol – simbol *Data Flow Diagram*

No	Nama	Gane dan Sarson	Yourdan dan De Marco	Keterangan
1.	Entitas Eksternal			Entitas eksternal, dapat berupa orang/unit terkait yang berintraksi dengan sistem.
2.	Proses			Orang atau unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
3.	Aliran Data			Aliran Data digunakan untuk menggambarkan 4.aliran data dari satu proses ke proses lainnya.
4.	Data Store			Penyimpanan data/file atau tempat data yang disimpan dalam proses

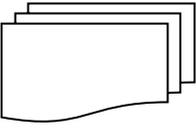
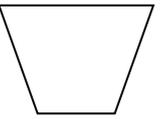
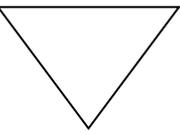
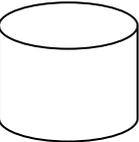
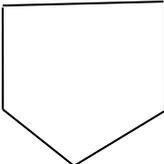
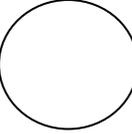
2.2.2 Block Chart

Kristanto (2008:75) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan

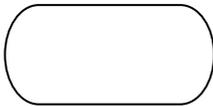
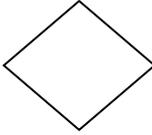
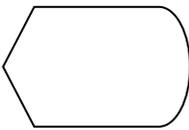
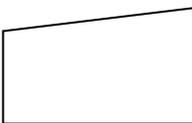


simbol-simbol tertentu. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.

Tabel 2.2 Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen
3.		Proses Manual
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (<i>Storage</i>)
7.		Proses apa saja yang tidak <i>terDefinisi</i> 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.

**Lanjutan Tabel 2.2** Simbol-simbol dalam *Block Chart*

10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukkan data secara manual.

2.2.3 Flow Chart

Ewolf (2012:16) mengemukakan, “*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya”.

Ewolf (2012:16) juga menjelaskan tentang simbol-simbol yang sering digunakan dalam *flowchart* dapat dilihat pada tabel berikut ini:

Tabel 2.3 Simbol-simbol dalam *Flow Chart*

NO	SIMBOL	ARTI
1.		Simbol <i>Start</i> atau <i>End</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.

Lanjutan Tabel 2.3 Simbol-simbol dalam *Flow Chart*

3.		Simbol <i>Input/Output</i> yang mendefinisikan masukan dan keluaran proses.
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.
5.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
7.		Simbol untuk menghubungkan antar proses atau antar symbol
8.		Inti keluaran, seperti layar monitor, <i>printer</i> , dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual
10.		Simbol yang menyatakan bagian dari program (subprogram)
11.		Simbol masukan atau keluaran dari atau ke sebuah pita <i>magnetic</i>
12.		Simbol <i>database</i> atau basis data



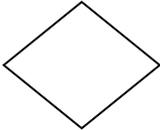
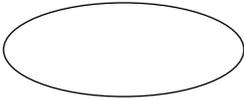
2.2.4 Entity Relationship Diagram (ERD)

Yakub (2012:60), “*Entity Relationship Diagram (ERD)* merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. Jenis-jenis *cordinalitas*:

1. *One to one (1:1)*
2. *One to many (1:M)*
3. *Many to many (M:M)*
4. *Many to one (M:1)*

Adapun simbol-simbol yang digunakan dalam *ERD*, yaitu:

Tabel 2.4 Simbol-simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1		Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
2		Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain; satu ke satu, satu ke banyak, dan banyak ke banyak.
3		Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.
4		Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasinya .



2.2.5 Kamus Data (*Data Dictionary*)

Kristanto (2008:72), “Kamus data adalah kumpulan elemen - elemen atau symbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap fiels atau file di dalam sistem”.

Bahra (2013:70),”Kamus data adalah katalog fakta tentang data dan kebutuhan”.

Kristanto (2008:72), menjelaskan simbol-simbol yang digunakan dalam kamus data, yaitu:

Tabel 2.5 Simbol-simbol dalam Kamus Data

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	Dan
3	[]	Memilih salah satu alternative
4	{ } ⁿ	n kali diulang/ bernilai banyak
5	()	data operasional
6	*...*	batas komentar

(Sumber : Kristanto 2008:72)

2.2.6 Pengertian Daftar Kejadian (*Event List*)

Kristanto (2008:70) menjelaskan bahwa "daftar kejadian (*event list*) digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan atau respon dari sistem”. Adapun cara mendeskripsikan daftar kejadian adalah sebgai berikut:

1. Pelaku adalah *entity* luar, jadi bukan sistem.
2. Menguji setiap *entity* luar dan mencoba mengevaluasi setiap *entity* luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tak sengaja menyatu dalam paket yang sama.
4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya interaksi normal antara sistem dengan *entity* luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.



5. Setiap aliran keluaran sebaiknya merupakan respon dari kejadian.
6. Setiap kejadian yang tidak berorientasi pada waktu dalam daftar kejadian sebaiknya mempunyai masukan sehingga sistem dapat mendeteksi kejadian yang berlangsung.
7. Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.

2.3 Teori Program

2.3.1 Pengertian Basis Data (*Database*)

Kadir (2013:15), “Database adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data”.

Yakub (2012:51), “Basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi). Relasi biasanya ditunjukkan dengan kunci (*key*) dari tiap file yang ada”.

Priyadi (2014:2) menyatakan, “Secara umum, pengertian basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital”.

Priyadi (2014:3) juga menjelaskan beberapa elemen basis data, diantaranya adalah sebagai berikut:

1. Tabel

Pada suatu basis data, tabel direpresentasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.

2. *Field*

Pada tabel basis data, kolom merupakan representasi untuk sebuah nama *field* yang pembacaan datanya dilakukan secara vertikal.

3. *Record*

Pada tabel basis data, baris merupakan suatu representasi untuk sebuah *record* yang pembacaan datanya dilakukan secara horizontal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel tersebut.



4. Kardinalitas

Kardinalitas merupakan batasan dari banyaknya hubungan, yang dapat dilakukan oleh himpunan entitas dalam melakukan relasi dengan himpunan entitas lainnya. Variasi kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri dari empat macam, yaitu:

- a. Satu ke satu \longrightarrow (1:1)
- b. Satu ke banyak \longrightarrow (1:N)
- c. Banyak ke satu \longrightarrow (N:1)
- d. Banyak ke banyak \longrightarrow (N:N)

2.3.2 Pengertian *HTML*

Asporudin (2013:44), “*Hyper Text Markup Language (HTML)* adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *webpage*”.

2.3.3 Pengertian *CSS*

Madcoms (2010:141), “*CSS (Cascading Style Sheets)* adalah suatu kumpulan kode-kode untuk memformat, yang mengendalikan tampilan isi dalam suatu halaman web”.

2.3.4 Pengertian Pemrograman *PHP (Personal Home Page)*

Anhar (2010:3), “*PHP* adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halamn itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script *PHP* dieksekusi pada server dimana *script* tersebut dijalankan”.

Menurut Wahana (2013:iii) “Pemrograman *PHP* adalah pemrograman berbasis web yang sudah sangat dikenal. Bahasa pemrograman *PHP* termasuk bahasa pemrograman berbasis web yang bersifat *cross platform* atau dapat dijalankan di berbagai macam sistem operasi”.

2.3.4.1 Konsep *PHP*

Rancangan dan konsep bahasa *PHP* sangat sederhana, bahkan lebih sederhana daripada *CGI (Common Gateway Interface)*. Cukup membuat sebuah



HTML biasa, dengan menambahkan kode – kode program yang diapit dalam tanda `<?.....?>`, maka jadilah skrip *PHP*. Persis seperti *javascript*, hanya berbeda pengapit dan dieksekusi di sisi *server*, sehingga saat kita membuka di *browser* kode *PHP* sudah diganti dengan hasil eksekusinya. Sementara itu, untuk membuat aplikasi CGI, kita harus mengerti dulu mekanisme HTTP dan protokolnya, apa itu *variable environment*, atau bagaimana mencetak *header* respon HTTP yang benar, sehingga cukup merepotkan.

2.3.4.2 Skrip *PHP*

Madcoms (2010:350) mengemukakan bahwa, “*Script PHP* tidak harus berdiri sendiri namun dapat disisipkan di antara kode *HTML*. *Script PHP* harus diawali dengan `<? Atau <?PHP` dan diakhiri dengan `?>`”.

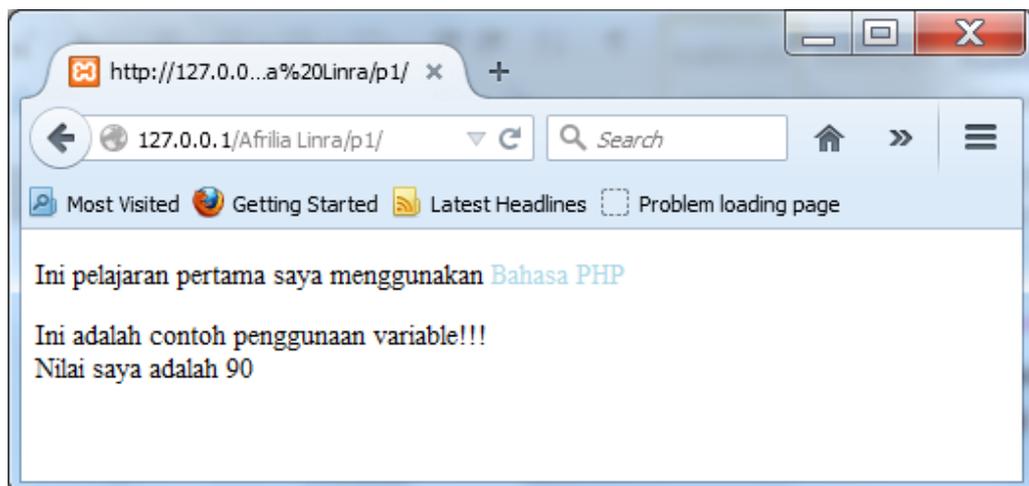
Berikut adalah contoh yang umum digunakan untuk menjelaskan tentang *PHP* sebagai *script* yang disisipkan (*embedded script*) dalam dokumen *HTML* :

```
<!doctype html>
<html lang="en">
<head>
    <title></title>
</head>
<body>
    <p>Ini pelajaran pertama saya
    <?PHP
        //kode yang digunakan untuk mencetak/ menampilkan data
        echo"menggunakan <font color='lightblue'>Bahasa PHP</font>";
    ?>
    </p>
    <p>
        Ini adalah contoh penggunaan variable!!!<br>
    <?PHP
        //pendefenisian data ke dalam variable
        $nilai=90;
        //menampilkan data yang ada di dalam variable
        echo "Nilai saya adalah $nilai";
```



```
?>
</p>
</body>
</html>
```

Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



Gambar 2.3 Tampilan *PHP* Sederhana Pada Browser

2.2.4.3 Teknik Penulisan Skrip *PHP*

Skrip *PHP* diawali dan diakhiri dengan menggunakan tag khusus. Ada 4 macam cara yang dapat digunakan untuk menuliskan script *PHP* di dalam suatu dokumen HTML. Adapun caranya sebagai berikut:

1. Cara 1

```
<PHP echo (“Script PHP”); ?>
```

Contoh:

```
<html>
<head>
<title> coba </title>
</head>
<body>
<?PHP
    echo “hai, saya dari script PHP ! “;
```



```
?>
</body>
</html>
```

2. Cara 2

```
<? Echo ("Script PHP"); ?>
```

Contoh:

```
<html>
<head>
<title> coba2 </title>
</head>
<body>
<?PHP
    echo "hai, saya dari script PHP ! ";
?>
</body>
</html>
```

3. Cara 3

```
<% echo("Script PHP"); %>
```

Contoh:

```
<html>
<head>
<title> coba3 </title>
</head>
<body>
<%
    echo "hai, saya dari script PHP ! ";
%>
</body>
</html>
```

4. Cara 4

```
<% =$namaVar; %>
```

Contoh:



```

<html>
<head>
<title> coba4 </title>
</head>
<body>
<%
    $str="hai, saya dari script PHP ! ";
%>
<%= $str; %>
</body>
</html>

```

2.3.4.4 Komentar

Komentar yang dimaksud dalam skrip *PHP* adalah bagian dari script *PHP* yang tidak akan dieksekusi, karena merupakan catatan fungsi dari script ayau menjelaskan maksud dari sebagian dari script yang dituliskan.

Ada 3 macam cara penulisan komentar dalam *PHP*:

1. `/* komentar */` : Cara penulisan komentar dari bahasa pemrograman C
2. `// komentar` : Cara penulisan komentar dari bahasa pemrograman C++
3. `# komentar` : Cara penulisan komentar dari pemrograman sript shell Bourne di Unix/Linux.

2.3.5 Pengertian *MySQL*

Kadir (2008:348), "*MySQL* adalah salah satu jenis database server yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses databasenya. Selain itu, ia bersifat open source pada berbagai platform".

Anhar (2010:21), "*MySQL (My Structure Query Language)* adalah sebuah perangkat lunak sistem manajemen basis data *SQL (Database Management System)* atau *DBMS* dari sekian banyak *DBMS*, seperti Oracle, MS *SQQL*, *Postrage SQL*, dan lain-lain".



Dari kedua pengertian di atas, dapat disimpulkan bahwa *MySQL* adalah jenis *software* pada *Relational Database Management System (RDBMS)* pada sebuah manajemen *database* sebagai basis data.

2.3.5.1 Fungsi-fungsi *MySQL*

Adapun fungsi-fungsi *MySQL* dapat dilihat pada tabel berikut ini:

Tabel 2.6 Fungsi-fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
1	<i>MySQL_connect()</i>	Koneksi ke <i>MySQL</i>	<i>mysql_connect (host, user, password)</i>
2	<i>MySQL_close()</i>	Menutup koneksi <i>MySQL</i>	<i>mysql_close (link_id)</i>
3	<i>MySQL_select_db()</i>	Memilih <i>database</i>	<i>mysql_select_db (database)</i>
4	<i>MySQL_query()</i>	Melakukan Query	<i>mysql_query (query)</i>
5	<i>MySQL_fetch_array()</i>	Mengambil hasil query dalam bentuk array asosiatif	<i>mysql_fetch_array (pengenal_hasil)</i>
6	<i>MySQL_num_rows()</i>	Jumlah data hasil query	<i>mysql_num_rows (pengenal_hasil)</i>
7	<i>MySQL_fetch_row()</i>	Mengambil hasil query perbaris dalam bentuk array	<i>mysql_fetch_row (pengenal_hasil)</i>
8	<i>MySQL_result</i>	Mengambil data hasil query	<i>mysql_query (query)</i>

(Sumber : Kristanto (2010:222))



2.3.6 Pengertian *Dreamweaver*

Menurut Madcoms (2008:1), “*Dreamweaver* adalah sebuah HTML editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web”.

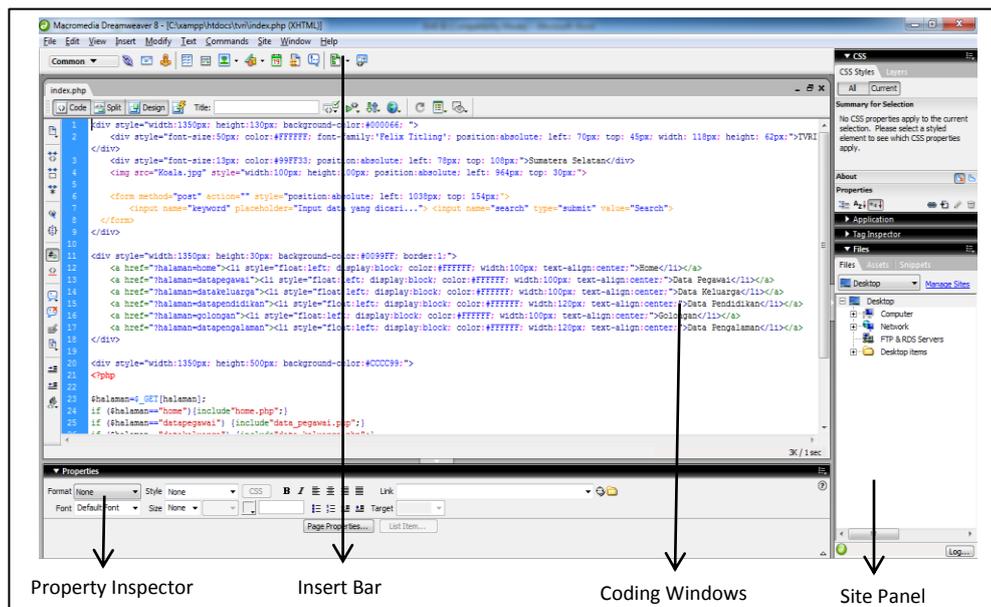
2.3.6.1 Komponen *Macromedia Dreamweaver*

Tampilan awal *Macromedia Dreamweaver* adalah sebagai berikut:



Gambar 2.4 Tampilan awal *Dreamweaver*

Kemudian berikut ini merupakan tampilan utama ruang kerja *Macromedia Dreamweaver*:



Gambar 2.5 Tampilan Ruang Kerja *Macromedia Dreamweaver*



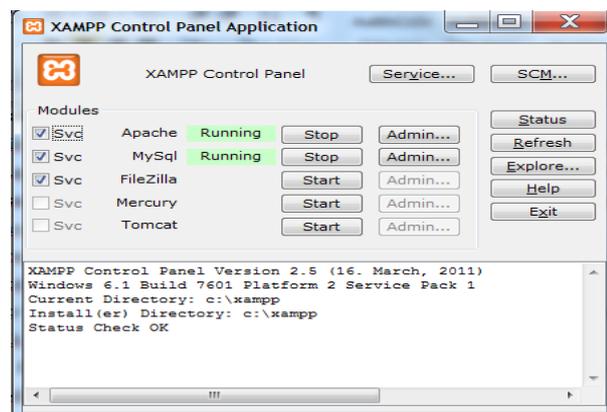
2.3.6.2 Sekilas mengenai XAMPP

Nugroho (2013:1), "XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web khususnya *PHP* dan *MySQL*."

Langkah-langkah pemasangan XAMPP dari file zip yang telah di download adalah sebagai berikut:

1. Unzip file XAMPP yang telah didownload, ke dalam file directori `c:/xampp`
2. Masuk ke dalam directori `c:/xampp`
3. Pilih file `xampp-control.exe`, kemudian jalankan dengan menekan `<enter>` atau klik dua kali, beri tanda cek pada baris *Svc Apache*, dialog konfirmasi untuk menjalankan Apache menjadi *Service*.
4. Klik tombol **OK**.
5. Setelah itu klik tombol start pada baris yang sama, menjalankan Apache.

Maka tampilan menjadi seperti berikut:



Gambar 2.6 Tampilan *Svc ApacheRunning*

2.3.7 JavaScript

Asropudin (2013:53), " *Javascript* adalah bahasa script *interpreter* untuk menciptakan isi yang aktif dalam halaman web".

Winarno, dkk (2014:134) menjelaskan bahwa cara menggunakan JavaScript adalah dengan menggunakan tag `<script>` dan `</script>`. JavaScript biasanya dituliskan pada bagian `<body>` dalam HTML. Contohnya:

```
<!Doctype html>
<html>
<body>
```



```
<script>
document.write("<h1>Ini adalah Heading </h1>");
document.write("<p>Ini adalah paragraf.</p>");
</body>
</html>
```

(Sumber : Winarno, dkk., 2014:134)