



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Irwansyah et al. (2014:02), “Komputer adalah perangkat elektronik yang beroperasi di bawah kendali instruksi yang memiliki memori, dapat menerima dan mengolah data sesuai dengan aturan yang ditetapkan, menghasilkan hasil atau informasi dan menyimpan hasil tersebut untuk digunakan di masa depan atau di kemudian hari.”

Erzha (2014:7), “Komputer adalah suatu alat untuk pengolahan informasi sesuai prosedur yang telah dirumuskan.”

Kesimpulannya, komputer adalah alat elektronik yang beroperasi di bawah kendali instruksi untuk pengolahan informasi sesuai prosedur yang telah dirumuskan.

##### 2.1.2. Pengertian Sistem

Kristanto (2011:1), “Sistem adalah kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan.”

Hutahaean (2014:2), “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu.”

Kesimpulannya, sistem adalah suatu kumpulan dari elemen atau prosedur yang saling terhubung dan bekerja sama untuk memroses masukan (*input*) yang secara bersama-sama mencapai tujuam tertentu.



### 2.1.3. Karakteristik Sistem

Sutabri (2012:20), menjelaskan karakteristik sistem adalah sebagai berikut:

- a. **Komponen Sistem (*Components*)**  
Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerjasama membentuk satu kesatuan.
- b. **Batasan Sistem (*boundary*)**  
Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.
- c. **Lingkungan Luar Sistem (*Environment*)**  
Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem disebut lingkungan luar sistem.
- d. **Penghubung Sistem (*interface*)**  
Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*.
- e. **Masukan Sistem (*Input*)**  
Energi yang di masukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*)
- f. **Keluaran Sistem (*Output*)**  
Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi.
- g. **Pengolah Sistem (*Process*)**  
Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi.
- h. **Sasaran Sistem (*Objective*)**  
Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

### 2.1.4. Klasifikasi Sistem

Sutabri (2012:22), menjelaskan sistem dapat diklasifikasikan menjadi beberapa bagian yaitu:

- a. **Sistem abstrak dan sistem fisik**  
Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia dan lain sebagainya.
- b. **Sistem alamiah dan sistem buatan manusia**  
Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia



merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine sistem*. Sistem informasi berbasis komputer merupakan contoh *human machinesistem* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

- c. Sistem determinasi dan sistem probabilistik  
Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program – program komputer yang dijalankan, sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistic*.

### 2.1.5. Prinsip Pengembangan Sistem

Jogianto (2005:38), sewaktu akan membangun atau mengembangkan sistem informasi, ada beberapa prinsip yang tidak boleh dilupakan yaitu sebagai berikut:

- a. Sistem yang dikembangkan adalah untuk manajemen.  
Setelah sistem selesai dikembangkan, maka yang akan menggunakan informasi dari sistem ini adalah manajemen, sehingga sistem harus dapat memenuhi kebutuhan manajemen.
- b. Sistem yang dikembangkan adalah investasi modal yang besar.  
Untuk pengembangan sistem membutuhkan dana yang tidak sedikit apalagi menggunakan teknologi yang mutakhir.
- c. Sistem yang dikembangkan memerlukan orang yang terdidik.  
Manusia lah yang berperan atas keberhasilan suatu sistem, baik dalam proses pengembangan, penerapan, maupun dalam operasinya.
- d. Tahapan kerja dan tugas yang harus dilakukan dalam pengembangan sistem.  
Proses pengembangan sistem umumnya melibatkan beberapa tahapan kerja dan melibatkan beberapa personal dalam bentuk suatu tim untuk mengerjakan.
- e. Proses pengembangan sistem yang tidak harus urut.  
Tahapan kerja pengembangan sistem merupakan langkah – langkah yang harus dilakukan. Langkah ini dapat saja tidak berurutan, tetapi dapat dilakukan secara bersama-sama.
- f. Jangan takut membatalkan proyek.  
Pada umumnya merupakan pantangan untuk membatalkan proyek yang sedang berjalan. Keputusan untuk meneruskan atau membatalkan suatu proyek memang harus didasarkan evaluasi yang cermat.
- g. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem.  
Kegagalan untuk membuat suatu dokumentasi kerja merupakan salah satu hal yang sering terjadi dan merupakan kesalahan kritis yang dibuat oleh analisis sistem.



### **2.1.6. Pengertian Informasi**

Djahir et al. (2014:10), “Informasi adalah hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan keputusan.”

Hutahaean (2014:9), “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya.”

Kesimpulannya, informasi adalah suatu data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya dan dapat digunakan sebagai alat bantu untuk pengambilan keputusan.

### **2.1.7. Pengertian Sistem Informasi**

Hutahaean (2014:13), “Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi yang menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.”

Menurut Rahmat dalam Djahir et al. (2014:14), “Sistem Informasi adalah kegiatan atau aktifitas yang melibatkan serangkaian proses, berisi informasi-informasi yang digunakan untuk mencapai tujuan.”

### **2.1.8. Pengertian Penjualan**

Indrajani (2015:62), menjelaskan “kegiatan penjualan terdiri atas penjualan barang dan jasa, baik secara kredit maupun secara tunai. Dalam transaksi penjualan kredit, jika pesanan dari pelanggan telah dipenuhi dengan pengiriman barang atau penyerahan jasa, untuk jangka waktu tertentu perusahaan memiliki piutang kepada pelanggannya. Dalam sistem penjualan secara tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli.”



### **2.1.9. Pengertian Pembayaran**

Menurut Permanasari dalam Hasibuan (2001:117), “Pembayaran adalah berpindahnya hak pemilikan atas sejumlah uang atau dan dari pembayar kepada penerimanya, baik langsung ataupun melalui media jasa- jasa perbankan.”

### **2.1.10. Pengertian Properti**

Surat Keputusan Menteri Perumahan Rakyat no.05/KPTS/BKP4N/1995, Ps 1:a:4, properti adalah tanah hak dan atau bangunan permanen yang menjadi objek pemilik dan pembangunan.

### **2.1.11. Pengertian Sistem Informasi Penjualan dan Pembayaran Properti Secara *Online* pada PT Raja Kapitan Jaya.**

Sistem Informasi Penjualan dan Pembayaran Properti Secara *Online* pada PT Raja Kapitan Jaya adalah suatu sistem yang berisi informasi-informasi penjualan properti dan terdapat proses pembayaran yang terkoneksi dengan internet sehingga memudahkan pembeli maupun pengembang untuk melakukan transaksi pembayaran.

## **2.2. Teori Khusus**

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

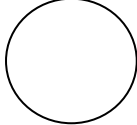
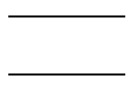

Sutabri (2012:117) menjelaskan “Data Flow Diagram adalah network yang menggambarkan suatu sistem automat/ komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.”

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. 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.



Rosa et al. (2013:71), menjelaskan “notasi pada DFD adalah sebagai berikut.”

**Tabel 2.1 Simbol-simbol *Data Flow Diagram***

No.	Simbol	Keterangan
1		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. Catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja
2		<i>File</i> atau basis data 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 (ERD)</i> , <i>Conceptual Data Model (CDM)</i> , <i>Physical Data Model (PDM)</i> ). Catatan : nama yang diberikan pada sebuah penyimpanan biasanya kata benda
3		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 sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. Catatan : nama yang digunakan pada masukan ( <i>input</i> ) atau keluaran ( <i>output</i> ) berupa kata benda.



No.	Simbol	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 :</p> <p>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>

(Sumber : Rosa et al. (2013:71))

Sutabri (2012:119) menjelaskan “tentang aturan main *Data Flow Diagram* yaitu:”

1. Di dalam *Data Flow Diagram* tidak boleh menghubungkan antara satu external *entity* dengan external *entity* lainnya secara langsung.
2. Di dalam *Data Flow Diagram* tidak boleh menghubungkan data store yang satu dengan data store lainnya secara langsung.
2. Di dalam *Data Flow Diagram* tidak boleh menghubungkan data store dengan external *entity* secara langsung.
4. Setiap proses harus ada memiliki *Data Flow* yang masuk dan juga *Data Flow* yang keluar.

Sutabri (2012:119) menjelaskan “tentang teknik membuat *Data Flow Diagram* yaitu:”

1. Mulai dari yang umum atau tingkatan yang lebih tinggi, kemudian diuraikan atau dijelaskan sampai yang lebih detail atau tingkatan yang lebih rendah, yang lebih dikenal dengan istilah TOP-DOWN ANALYSIS.
2. Jabarkan proses yang terjadi di dalam data flow diagram sedetail mungkin sampai tidak dapat diuraikan lagi.
2. Peliharalah konsistensi proses yang terjadi di dalam DFD, mulai dari diagram yang tingkatannya lebih tinggi sampai dengan diagram yang tingkatannya lebih rendah.
4. Berikan label yang bermakna untuk setiap symbol yang digunakan seperti:
  - a. Nama yang jelas untuk external *entity*.



- b. Nama yang jelas untuk proses.
- c. Nama yang jelas untuk data flow.
- d. Nama yang jelas untuk data store.

Rosa et al. (2013:72) menjelaskan tentang “tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:”

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

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

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem 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 kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, 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 2.

### 2.2.2. Pengertian Blockchart



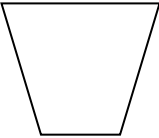
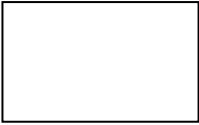
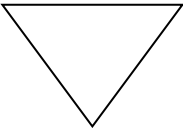
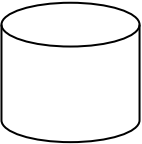
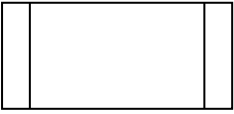
Kristanto (2011:68) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”



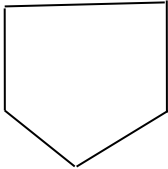
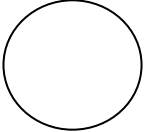

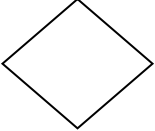




Kristanto (2008:75) juga menjelaskan tentang “simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini.”

**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
2.		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 terdefinisi termasuk aktivitas fisik.



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

(Sumber : Kristanto (2011:68))

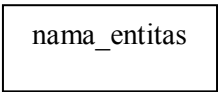
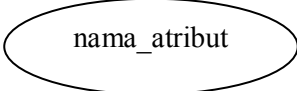
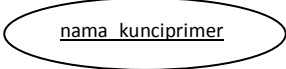
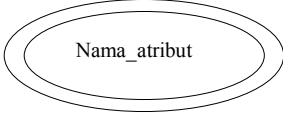
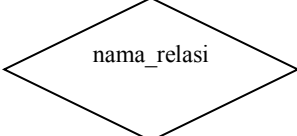
### 2.2.3. Pengertian ERD (*Entity Relational Diagram*)

Rosa et al. (2013:50), “ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional.”




Rosa et al. (2013:50), menjelaskan “simbol-simbol yang digunakan dalam ERD, yaitu:”

**Tabel 2.3 Simbol-simbol *Entity Relationship Diagram* (ERD)**

No	Simbol	Keterangan
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 komputer; 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
2.	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 multivalu / <i>multivalve</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
4.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.



No	Simbol	Keterangan
5.	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. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.




(Sumber : Rosa et al. (2008 :75))

#### 2.2.4. Pengertian *Flowchart*


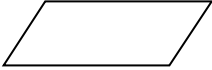
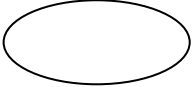
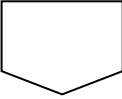

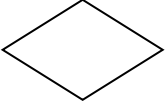
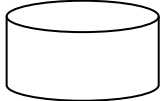


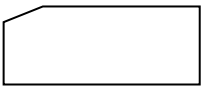
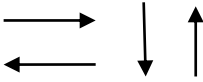


Siallagan (2009:6), "*Flow chart* adalah suatu diagram alir yang mempergunakan simbol atau tanda untuk menyelesaikan masalah."

Fungsi *Flowchart* adalah untuk mengetahui alur sistem yang akan dilakukan oleh *user* dalam hal yang bertindak sebagai *user* adalah admin, karyawan dan pimpinan perusahaan sehingga kita dapat mengetahui proses yang dilakukan admin untuk memasukkan data rencana kerja perusahaan dan proses yang dilakukan karyawan dan pimpinan untuk mendapatkan data rencana kerja perusahaan pada sistem informasi yang telah kami buat.

**Tabel 2.4 Simbol-simbol dalam *Flow Chart***

No.	Simbol	Keterangan
1.		Terminal menyatakan awal atau akhir dari suatu algoritma.
2.		Menyatakan proses.
2.		Proses yang terdefinisi atau sub program.



No	Simbol	Keterangan
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Menyatakan masukan dan keluaran ( <i>input/output</i> ).
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.
12.		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).

(Sumber : Siallagan (2009:6))



### 2.2.5. Pengertian Kamus Data

Rosa et al. (2013:73), “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).”

Fungsi kamus data adalah untuk mengetahui *field* dari tabel-tabel yang ada di dalam database secara lebih detail termasuk tipe *field* dan panjang karakter tulisan yang ada pada masing-masing *field*.

Rosa et al. (2013:74), menjelaskan “simbol-simbol yang di gunakan dalam kamus data, yaitu :”

**Table 2.5 Simbol-simbol dalam Kamus Data**

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

(Sumber : Rosa et al. (2013:74))

## 2.3. Teori Program

### 2.3.1. Pengertian Basis Data (*Database*)

Tim EMS (2014:54), “Basis data adalah tempat untuk menyimpan data dan nantinya data ini bias diambil lagi.”

Yanto (2016:14), “Basis data sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar *record*.”

Prasatio (2014:45), “Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.”



### 2.3.2. Pengertian *MySQL*

Prasetio (2014:29), “*MySQL* adalah sebuah *database*, dan dapat anda bayangkan seperti sebuah tempat penyimpanan.”

### 2.3.3. Pengertian *HTML*

Kadir (2013:20), “*HTML (Hypertext Markup Language)* adalah bahasa skrip yang digunakan untuk menyusun halaman *web*.”

Tim EMS (2014:2), “ *HTML* adalah singkatan dari *Hypertext Markup Language*. Bahasa pemrograman ini terdiri dari *tag* dan aturan-aturan yang memungkinkan membuat dokumen *hypertext*.”

Prasetio (2014:3), “*HTML* adalah bahasa pemrograman yang digunakan untuk mendesain sebuah halaman *web*.”

### 2.3.4. Sekilas Tentang *PHP*

Utomo (2014:2), “*PHP* dirancang oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1994, *PHP* digunakan untuk mendeteksi *user* yang berkunjung pada situs. Selain sebagai bahasa pemrograman *web* yang dapat dieksekusi sendiri, *PHP* juga dikenal sebagai *embedded language*. Artinya, anda dapat memasukkan kode *PHP* ke dalam bahasa *HTML* yang sebelumnya dikenal sebagai bahasa pembentuk halaman *website*.”

#### 2.3.4.1. Pengertian *PHP (Hypertext Preprocessor)*

Tim EMS (2014:56), “*PHP* adalah singkatan dari *PHP Hypertext Preprocessor* merupakan bahasa scripting seperti *HTML* yang lazim digunakan di halaman *web*.”

Badiyanto (2013:32), “*PHP (Hypertext Preprocessor)* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML/ PHP* banyak dipakai untuk membuat situs *web* dinamis.”



### 2.3.4.2. Tipe data PHP

Tim EMS (2014:72) menjelaskan tipe data yang dapat diakomodasi di PHP, yaitu:

1. Integer  
Integer adalah tipe data paling standar, ini hanya menyatakan kebenaran, apakah TRUE (benar) atau FALSE (salah). Untuk menyatakan sebuah boolean, anda bisa menggunakan kata kunci TRUE atau FALSE.
2. Integer  
Sebuah integer adalah bilangan bulat (bukan pecahan), baik negatif atau positif. Misalnya {...,-2, -1, 0, 1, 2, ...}.  
Integer bisa dituliskan dalam satuan desimal (berbasis 10), heksadesimal (berbasis 16), atau oktal (berbasis 8) dan bisa juga ditambahi tanda plus atau minus (- atau +) tapi penandaan ini opsional saja.
3. Floating Point  
Floating point atau nomor pecahan atau juga bilangan real, bisa didefinisikan dengan syntax berikut ini :  

```
<?php
$a = 2.652;
$b = 2.2.e3;
$c = 7E - 10;
?>
```
4. String  
String adalah untaian karakter, sehingga karakter sama seperti byte. Ada 256 karakter yang bisa dijadikan string. Sebuah string bisa didefinisikan dengan berbagai cara, tapi yang paling mudah adalah dengan satu petik.
5. Null  
Nilai NULL mempresentasikan variabel tanpa value. NULL adalah nilai yang mungkin untuk NULL. Sebuah variabel akan menjadi null jika :
  - a. Di alokasikan konstanta NULL
  - b. Variabel yang belum diset ke nilai apapun.
  - c. Menerima unset().

### 2.3.4.3. Skrip PHP

Utomo (2014:18), “PHP merupakan *embedded language* pada HTML. PHP dapat dimasukkan pada kode HTML dalam satu *file*.” Berikut ini contoh sintaks dasar penggunaan PHP:

```
<!doctype html>
<HTML>
<HEAD>
<META charset="utf-8">
<TITLE>Testing</TITLE>
```



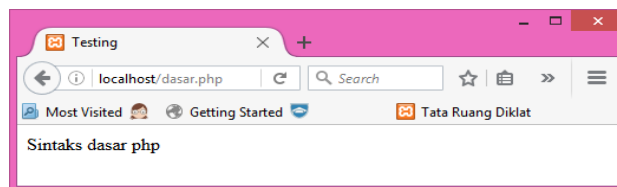


```

</HEAD>
    <?php
    Echo "Sintaks dasar php";
    ?>
<BODY>
</BODY>
</HTML>

```

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



**Gambar 2.1.** Tampilan PHP Sederhana

### 2.3.5. Hubungan antara HTML, PHP, dan MySQL

Ketika *user* akan mengakses data perusahaan maka *PHP* akan berusaha untuk mengambil data dari *MySQL* menggunakan salah satu fungsi yang ada di *PHP* yaitu *mysql\_query()*. Setelah data diambil, maka *PHP* akan mengirimkan hasilnya ke *webserver* yang berbasis *HTML* dengan menggunakan salah satu fungsi *PHP* yaitu *echo()*. Selanjutnya *HTML* bertugas untuk menampilkan halaman *web* ke *browser* sesuai kebutuhan menggunakan tag-tag *HTML*.

### 2.3.6. Java Script

Wahana Komputer (2012:2), “*Java Script* adalah salah satu bahasa *script website* yang paling banyak digunakan untuk menambah manipulasi *script HTML* dan *CSS* pada sisi *client/browser*.”

Javascript merupakan bahasa pemrograman client side. Kode Javascript dituliskan dalam bentuk fungsi yang ditaruh di tag `<head>` yang dibuka dengan tag `<script type="teks/javascript" src="alamat.js">` dan diakhir dengan tag `</script>`.



### 2.3.6.1. Variabel dan Operator JavaScript

Wahana Komputer (2012:2) menjelaskan variabel dan operator yang biasa digunakan *Java Script*:

#### 1. Variabel

Wahana Komputer (2012:48), “Variabel adalah sebuah *identifier*/ pengenalan/ nama yang mempunyai nilai yang dapat berubah-ubah. Variabel merupakan bagian dalam sebuah bahasa pemrograman yang digunakan untuk menyimpan data sementara waktu.”

Variabel dapat dideklarasikan dengan *keyword var* atau tanpa *keyword*. Pendeklarasian variabel dengan *keyword var* pada suatu blok akan menyebabkan *variabel* itu hanya dapat diakses pada blok itu atau disebut variabel lokal. Sedangkan pendeklarasian variabel tanpa *var* dimana pun.

#### 2. Operator

Operator adalah simbol yang digunakan untuk melakukan operasi dua nilai atau lebih.

- a. Operator *Assignment* adalah operator yang digunakan untuk memberikan nilai pada suatu variabel. Operator *assignment* asli disimbolkan dengan karakter = (sama dengan). Operator tersebut dapat digabungkan dengan operator aritmatik sehingga dapat melakukan operasi *arithmetic*.
- b. Operator *Aritmatic* adalah operator yang digunakan untuk melakukan operasi matematika, baik penjumlahan, pengurangan, perkalian, pembagian, dan modulo.
- c. Operator Logika adalah operator yang digunakan untuk melakukan operasi *boolean* (*true* dan *false*) atau (1 dan 0). Pada *Java Script* terdapat 3 buah operator logika, yaitu *AND* (&& dan &), *OR* (|| dan |), dan *NOT* (!).
- d. Operator *Conditional* adalah operator yang mempunyai 3 buah ekspresi. Ekspresi pertama adalah nilai *boolean*. Ekspresi kedua dijalankan apabila ekspresi pertama bernilai *true*. Ekspresi ketiga akan dijalankan apabila ekspresi pertama bernilai *false*.
- e. Operator *String* adalah operator yang digunakan untuk melakukan penggabungan *string*. Hanya ada satu operator, yaitu +. Apabila terjadi operasi antara tipe *int* dengan *string* maka dilakukan operasi string dan hasilnya dalam bentuk *string*.
- f. Operator Perbandingan adalah operator yang digunakan untuk membandingkan sebuah nilai. Jika dua buah nilai memenuhi operasi perbandingan maka akan menghasilkan nilai *true*. Sebaliknya, jika tidak memenuhi akan menghasilkan nilai *false*.