

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Pengertian Nilai**

Kimball Young Mengemukakan nilai adalah asumsi yang abstrak dan sering tidak disadari tentang apa yang dianggap penting dalam masyarakat.

A.W.Green Nilai adalah kesadaran yang secara relatif berlangsung disertai emosi terhadap objek. Woods Mengemukakan bahwa nilai merupakan petunjuk umum yang telah berlangsung lama serta mengarahkan tingkah laku dan kepuasan dalam kehidupan sehari-hari M.Z.Lawang Menyatakan nilai adalah gambaran mengenai apa yang diinginkan, yang pantas, berharga, dan dapat memengaruhi perilaku sosial dari orang yang bernilai tersebut.

Hendropuspito Menyatakan nilai adalah segala sesuatu yang dihargai masyarakat karena mempunyai daya guna fungsional bagi perkembangan kehidupan manusia.

Karel J. Veeger Menyatakan sosiologi memandang nilai-nilai sebagai pengertian-pengertian (sesuatu di dalam kepala orang) tentang baik tidaknya perbuatan-perbuatan. Dengan kata lain, nilai adalah hasil penilaian atau pertimbangan moral.

#### **3.2 Informasi**

Informasi adalah pesan (ucapan atau ekspresi) atau kumpulan pesan yang terdiri dari order sekuens dari simbol, atau makna yang dapat

ditafsirkan dari pesan atau kumpulan pesan. Informasi dapat direkam atau ditransmisikan. Informasi adalah data yang telah diberi makna melalui konteks. Sebagai contoh, dokumen berbentuk spreadsheet (semisal dari Microsoft Excel) seringkali digunakan untuk membuat informasi dari data yang ada di dalamnya. Laporan laba rugi dan neraca merupakan bentuk informasi, sementara angka-angka di dalamnya merupakan data yang telah diberi konteks sehingga menjadi punya makna dan manfaat.

### **3.3 *Internet***

Pengertian *Internet* adalah hubungan banyak komputer dengan berbagai tipe yang membentuk sistem jaringan global yang terbentuk dari jaringan komputer lokal maupun regional yang memungkinkan adanya komunikasi data antar komputer tersebut melalui jaringan telepon.

Agar komputer-komputer dari suatu jaringan dapat berkomunikasi maka diperlukan suatu *Protocol* atau aturan standar komunikasi baik antar komputer maupun antar jaringan komputer. *Protocol* yang umum digunakan untuk jaringan komputer dan menjadi standar *Protocol* yang digunakan pada jaringan *internet* adalah *Transmission Control Protocol / Internet Protocol* (TCP / IP). *Transmission Control Protocol* (TCP) yang memastikan bahwa semua bekerja dengan benar, sedangkan *Internet Protocol* (IP) yang mentransmisikan data dari satu komputer ke komputer lain. Untuk dapat mengakses *internet* harus berlangganan salah satu *Internet Service Provider* (ISP) atau disebut dengan jasa penyedia *internet*.

### 3.4 Sistem

Sistem berasal dari bahasa Latin (*syst ma*) dan bahasa Yunani (*sust ma*) adalah sekumpulan unsur / elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan.

Definisi sistem menurut para ahli :

#### 1. Ludwig Von Bertalanfy

Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.

#### 2. Anatol Rapoport

Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

#### 3. L. Ackoff

Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

### 3.5 Website

*Website* atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan. Sedangkan *website* dinamis adalah seperti *Facebook*, *Twitter*, dan lain-lain. Dalam sisi pengembangannya, *website* statis hanya bisa di-*update* oleh pemiliknya saja, sedangkan *website* dinamis bisa di-*update* oleh pengguna maupun pemilik.

### 3.6 XAMPP

*XAMPP* adalah sebuah *software web server apache* yang didalamnya tersedia *database server mysql* dan *support php programming*. *XAMPP* merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah setiap menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support (PHP 4 dan PHP 5)* dan beberapa modul lainnya. Hanya bedanya pada versi untuk *Windows* sudah dalam bentuk instalasi grafis dan yang *Linux* dalam bentuk *file* terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk *Windows* adalah memiliki fitur untuk mengaktifkan sebuah *server* secara grafis, sedangkan *Linux* masih berupa perintah-perintah di dalam *console*. Oleh karena itu yang versi untuk *Linux* sulit untuk dioperasikan. Dulu *XAMPP* untuk *Linux* dinamakan *LAMPP*, sekarang diganti namanya menjadi *XAMPP FOR LINUX*.

### 3.7 PHP

*PHP* adalah bahasa pemrograman *server side* yang sudah banyak digunakan pada saat ini, terutama untuk pembuatan *website* dinamis. Untuk hal-hal tertentu dalam pembuatan *web*, bahasa pemrograman *PHP* memang diperlukan, misalnya saja untuk memproses data yang dikirimkan oleh pengunjung *web*.

*PHP* pertama kali dibuat oleh *Rasmus Lerdorf* pada tahun 1995. Pada waktu itu *PHP* bernama *FI (Form Interpreted)*. Pada saat tersebut *PHP* adalah sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*.

*Web Server* yang mendukung *php* dapat ditemukan dimana-mana dari mulai *IIS* sampai dengan *apache*, dengan konfigurasi yang relatif mudah. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan. Dalam sisi pemahaman, *php* adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.

*PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui konsol serta juga dapat menjalankan perintah-perintah sistem.

Apabila akan membuat *script* dasar *PHP*, maka *script* tersebut harus ditulis di antara tanda berikut ini:

```
<?php  
.  
.  
?>
```

```
atau
<?
.
.
?>
```

Tanda `<?php` atau `<?` dinamakan *tag* pembuka dari *script*, sementara `?>` disebut juga dengan *tag* penutup. Kedua tanda itu mutlak harus ada karena berfungsi sebagai penanda bahwa *script* tersebut merupakan *script* dasar *PHP* dan nantinya akan diproses oleh *server*.

Contoh *script PHP* sederhana :

```
<?php
echo "Hello Friend!!";
?>
```

### 3.8 CSS

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa *file*. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*.

Misalkan akan mengatur agar *tag* h1, h2 sampai h6 semua menggunakan warna merah, maka kode CSS nya menjadi :

```
h1,h2,h3,h4,h5,h6 { color: red }
```

Perhatikan penulisan h1,h2,h3,h4,h5,h6 dipisahkan oleh koma.

```

<html>
<head>
<title>Contoh Selector CSS</title>
<style type="text/css">h1,h2,h3,h4,h5,h6{color: red}</style>
</head>
<body>
<h1>Text Ini Berwarna Merah</h1>
<h2>Text Ini Berwarna Merah</h2>
<h3>Text Ini Berwarna Merah</h3>
<h4>Text Ini Berwarna Merah</h4>
<h5>Text Ini Berwarna Merah</h5>
<h6>Text Ini Berwarna Merah</h6>
</body>
</html>

```

Jika diterjemahkan ke kalimat bahasa Indonesia maksudnya adalah untuk mengatur warna dari h1 sampai h6 ke warna merah (*red*).

### 3.9 *Java Script*

*JavaScript* diperkenalkan pertama kali oleh *Netscape* pada tahun 1995. Pada awalnya bahasa yang sekarang disebut *JavaScript* ini dulunya dinamai "*LiveScript*" yang berfungsi sebagai bahasa sederhana untuk *browser Netscape Navigator 2* yang sangat populer pada saat itu. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan *Sun* (pengembang bahasa pemrograman "*Java*") pada masa itu, maka *Netscape* memberikan nama "*JavaScript*" kepada bahasa tersebut pada tanggal 4 Desember 1995. Pada saat yang bersamaan *Microsoft* sendiri mencoba untuk mengadaptasikan teknologi ini yang disebut sebagai "*Jscript*" di *browser Internet Explorer 3*. *JavaScript* sendiri merupakan modifikasi dari bahasa pemrograman *C++* dengan pola penulisan yang lebih sederhana dari bahasa pemrograman *C++*.

*JavaScript* adalah bahasa pemrograman yang khusus untuk halaman *web* agar halaman *web* menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu *Java* dan *Script*. *Java* adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program.

Ada beberapa hal yang harus diperhatikan dalam pengelolaan pemrograman *JavaScript*, diantaranya *JavaScript* adalah “*case sensitive*”, yang artinya *JavaScript* membedakan huruf besar dan huruf kecil, Jika pernah belajar bahasa pemrograman seperti *Turbo C* atau *C++*, maka sama seperti bahasa pemrograman tersebut, dimana huruf T tidak sama dengan huruf t. Dalam bahasa pemrograman *JavaScript* juga, sebagai contoh fungsi perintah *var* tidak boleh ditulis *Var* dan juga tidak boleh ditulis *VAR* (huruf besar semua), yang benar adalah *var* (huruf kecil semua). Perintah lain adalah *new Date* tidak boleh ditulis *new date* (huruf kecil semua), dan banyak yang lainnya.

Teknik penulisan *JavaScript* terbagi menjadi dua, yaitu penulisan secara langsung dan penulisan secara terpisah (*external JavaScript*).

- a. Penulisan *JavaScript* secara langsung, merupakan teknik yang menuliskan kode *script* dan dokumen *HTML* dalam satu halaman, seperti pada *script* menampilkan hari dan tanggal.

```
<html>
<head>
<script type="text/javascript">
document.write("hello blogger");
</script>
</head>
<body>
Contoh JavaScript Internal
```

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

Hasilnya adalah sebagai berikut :

### Contoh JavaScript Internal

- b. Penulisan *JavaScript* secara terpisah (*External JavaScript*), merupakan teknik penulisan bahasa *script* antara dokumen *HTML* dan *JavaScript* dengan cara dipisahkan, seperti pada *script mouse cursor sparkles*. Jika akan menjalankan *JavaScript* yang sama dalam beberapa kali pada halaman yang berbeda, tetapi tidak mau disibukkan harus menulis ulang *script* yang diinginkan disetiap halaman. Jadi, antara dokumen *HTML* dan *JavaScript* dipisahkan, kemudian *file* tersebut dipanggil dari dokumen *HTML*. File *JavaScript* disimpan dengan ekstensi *.js*.

```
<html>
<head>
<script src="myscript.js"
type="text/javascript"></script>
</head>
<body>
Contoh JavaScript External
</body>
</html>
```

Isi dari *myscript.js* adalah sebagai berikut :

```
document.write("hello blogger");
```

Hasilnya adalah sebagai berikut :

### Contoh JavaScript External

### 3.10 MySQL

*MySQL* adalah sebuah perangkat lunak pembuat *database* yang bersifat terbuka atau *open source* dan berjalan disemua *platform* baik *Linux* maupun *Windows*. *MySQL* merupakan program pengakses *database* yang bersifat *network* sehingga dapat digunakan untuk aplikasi *Multi User* (Pengguna Banyak).

*MySQL* adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial.

*MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*.

### 3.11 Notepad++

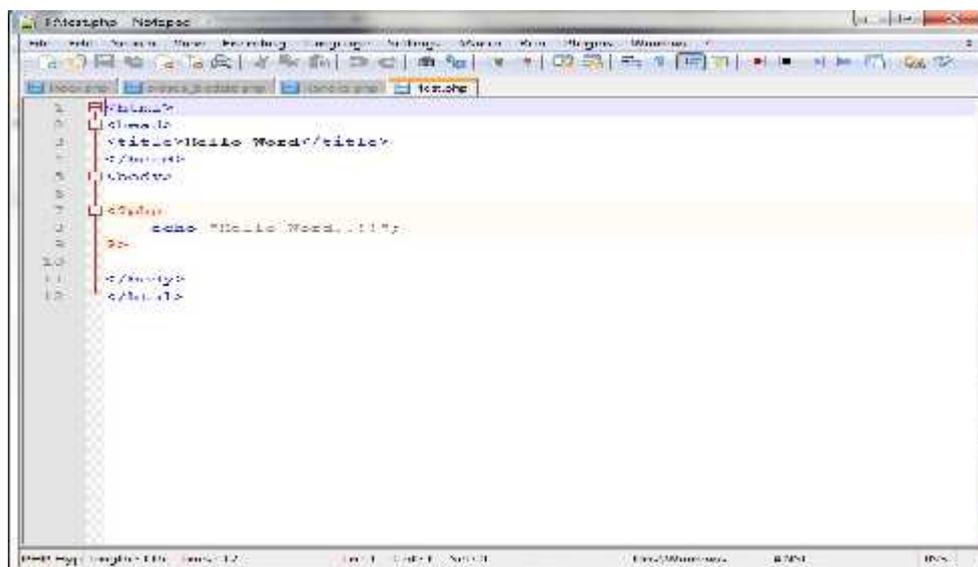
*Notepad++* adalah sebuah penyunting teks dan penyunting kode sumber yang berjalan di sistem operasi *Windows*. *Notepad++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai *bahasa* pemrograman.

*Notepad++* didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh *Sourceforge.net* dengan telah diunduh lebih dari 27 juta

kali dan dua kali memenangkan penghargaan *SourceForge Community Choice Award for Best Developer Tool*.

Aplikasi ini juga mendukung beberapa bahasa pemrograman berbasis *Windows* antara lain:

- C
- C++
- Java
- C#
- XML
- HTML
- PHP
- Javascript
- RC File
- Makefile
- nfo
- doxygen
- file ini
- batch
- ASP
- VB/VBS
- SQL
- Objective-C
- CSS
- Pascal
- Perl
- Python
- Lua
- Unix Shell Script
- Fortran
- NSIS
- Flash Action Script



Gambar 3.1 Notepad++

Tabel 3.1 fungsi toolbar notepad++

No	Toolbar	Fungsi
1		Untuk membuat <i>document</i> baru
2		Untuk membuka <i>document</i> yang sudah ada
3		Untuk menyimpan <i>document</i>
4		Untuk menyimpan semua <i>document</i>
5		Untuk menutup <i>document</i>
6		Untuk menutup semua <i>document</i>
7		Untuk mencetak <i>document</i>
8		Untuk memindahkan teks/ <i>script</i>
9		Untuk menyalin teks/ <i>script</i>
10		Untuk menempelkan teks/ <i>script</i> yang sudah di salin
11		Untuk menampilkan kembali kondisi sebelumnya
12		Untuk menampilkan kembali kondisi sesudahnya

No	Toolbar	Fungsi
13		Untuk mencari teks/ <i>script</i>
14		Untuk menghapus teks/ <i>script</i> yang sudah dicari
15		Untuk memperbesar tampilan <i>document</i>
16		Untuk meperkecil tampilan <i>document</i>
17		Untuk mensinkronkan guliran vertikal
18		Untuk mensinkronkan guliran horizontal
19		Untuk menampilkan penggal baris
20		Untuk menampilkan semua karakter
21		Untuk menampilkan panduan identasi
22		Untuk dialog definisi pengguna
23		Untuk merekam makro
24		Untuk memberhentikan rekaman makro
25		Untuk memutar rekaman makro
26		Untuk menjalankan makro beberapa kali
27		Untuk menyimpan rekaman makro
28		Untuk menampilkan <i>nppFTP window</i>
29		Untuk mengaktifkan <i>spelling</i> dan <i>checker</i> pada teks/ <i>script</i>

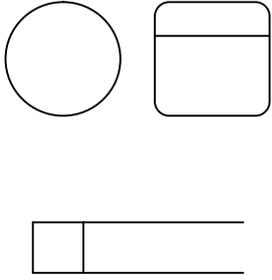
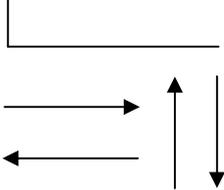
### 3.12 Diagram Alir Data

Diagram Alir Data (DAD) atau *Data Flow Diagram (DFD)* adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. *DFD* merupakan alat bantu dalam menggambarkan atau menjelaskan, *DFD* ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi.

Tujuan membuat Diagram Alir Data adalah :

- a. Merancang sebuah algoritma awal suatu program.
- b. Untuk dapat menunjukkan secara jelas arus pengendalian suatu algoritma dengan cara melaksanakan suatu rangkaian kegiatan secara logis dan sistematis.
- c. Dapat memberikan gambaran dua dimensi yang berupa simbol-simbol garis, dimana masing-masing simbol mempunyai arti dan fungsinya sendiri.
- d. Simbol tersebut dipakai untuk menunjukkan berbagai kegiatan operasi dan jalur pengendalian.

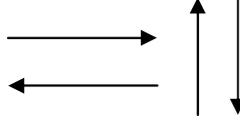
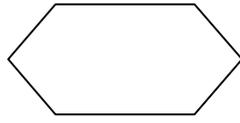
Tabel 3.2 Keterangan Simbol Diagram Alir Data

Simbol	Keterangan
	<p>Process (Proses), merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan.</p>
	<p><i>Data Store</i> (Simpan Data), dapat berupa suatu <i>file</i> atau <i>database</i> pada sistem komputer atau catatan manual.</p>
	<p><i>External Entity</i> (Entitas Eksternal), merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain.</p>

### 3.13 Flowchart

*Flowchart* (Bagan Alir) adalah bagan (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi

Tabel 3.3 Keterangan Simbol *Flowchart* Program

Simbol	Keterangan
 Terminal	Simbol untuk awal (permulaan) dan akhir dari suatu program.
 Proses	Simbol yang memproses proses atau pernyataan pada program.
 Garis Alur	Simbol yang menggabungkan dari simbol satu ke simbol yang lain.
 Persiapan	Simbol pemilihan untuk mempresentasikan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam <i>storage</i> (penyimpanan).
 Input / Output	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
 Konektor / Penghubung	Simbol untuk keluar masuk atau penyambungan proses dalam lembar atau halaman yang sama.

### 3.14 *Entity Relationship Diagram*

*Entity Relationship Diagram (ERD)* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

*Entity Relationship Diagram* digunakan untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

#### a. *Entity*

*Entity* merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari *entity* ini biasanya digambarkan dengan persegi panjang. *Entity* diberi nama dengan kata benda dan dapat dikelompokkan sebagai berikut :

- 1) Nama Orang
- 2) Nama Benda
- 3) Nama Lokasi
- 4) Nama Kejadian (terdapat unsur waktu didalamnya)

#### b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips*. Ada dua jenis atribut, yaitu :

- 1) *Identifier (key)* digunakan untuk menentukan suatu entity secara unik (*primary key*).
- 2) *Descriptor (nonkey attribute)* digunakan untuk menspesifikasikan karakteristik dari suatu entity yang tidak unik.

c. Hubungan atau Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

1) Satu ke Satu (*One to One*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.



Gambar 3.2 Hubungan Relasi Satu ke Satu

2) Satu ke Banyak (*One to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.



Gambar 3.3 Hubungan Relasi Satu ke Banyak

### 3) Banyak ke Banyak (*Many to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B



Gambar 3.4 Hubungan Relasi Banyak ke Banyak