

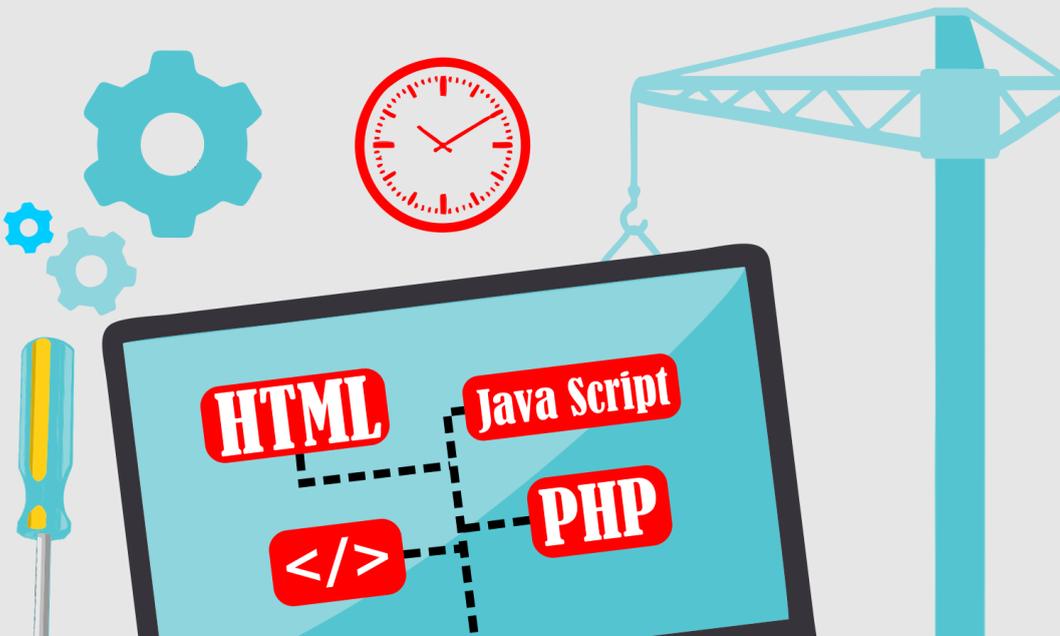
JILID-1

# PEMROGRAMAN WEB

Endah Ratna Arumi, S.Kom., M.Sc

Setiya Nugroho, S.Kom., M.Eng

Pristi Sukmasetya, S.Kom., M.Kom



# **PEMROGRAMAN WEB**

**JILID - 1**

Endah Ratna Arumi, S.Kom., M.Cs.

Setiya Nugroho, S.Kom., M.Eng.

Pristi Sukmasetya, S.Komp., M.Kom.

# PEMROGRAMAN WEB

## JILID - 1

**Perpustakaan Nasional: Katalog Dalam Terbitan (KDT)**

ISBN: 978-623-7261-23-0

viii, 116 hlm, uk. 15.5x23 cm

Hak Cipta 2020 pada Penulis

Hak penerbitan pada UNIMMA PRESS. Bagi mereka yang ingin memperbanyak sebagian isi buku ini dalam bentuk atau cara apa pun harus mendapatkan izin tertulis dari penulis dan penerbit UNIMMA PRESS.

### **Penulis:**

Endah Ratna Arumi, S.Kom., M.Cs.

Setiya Nugroho, S.Kom., M.Eng.

Pristi Sukmasetya, S.Komp., M.Kom.

### **Editor:**

Agus Setiawan, M.Eng.

### **Layout**

Muhammad Latifur Rochman, A.Md.

### **Desain sampul:**

Muhammad Latifur Rochman, A.Md.



Penerbit:

**UNIMMA PRESS**

**Anggota APPTI Nomor 003.083.1.02.2019**

Gedung Rektorat Lt. 3 Kampus 2 Universitas Muhammadiyah Magelang

Jl. Mayjend. Bambang Soengeng, Mertoyudan, Magelang 56172

Telp. (0293) 326945

E-Mail: [unimmapress@ummgl.ac.id](mailto:unimmapress@ummgl.ac.id)

Publikasi: [ebook.ummgl.ac.id](http://ebook.ummgl.ac.id)

Hak Cipta dilindungi Undang-undang

All Right Reserved

Cetakan I, Desember 2020

# KATA PENGANTAR

Mahasiswa Informatika harus mendapatkan pengetahuan teoretis dan praktik pemrograman dalam beberapa paradigma agar sudut pandang mahasiswa tidak sempit. Pemrograman fungsional merupakan salah satu paradigma yang wajib diberikan. Karena ada kaitannya dengan pengajaran pemrograman pada paradigma lain, buku ajar Pemrograman Web Jilid 1 ini menunjang perkuliahan pemrograman program studi Informatika. Beberapa bagian yang dipandang terlalu sulit untuk dibahas karena kurangnya waktu, dapat dibahas pada pelajaran *web* dinamis dengan paradigma yang berikutnya diajarkan.

Buku ajar ini dirancang bukan merupakan *textbook*, melainkan sebagai buku kerja mahasiswa, yang digunakan untuk membantu mahasiswa dalam memahami dan berlatih merancang, mengonstruksi, menulis dan membaca program dalam notasi fungsional. Beberapa contoh akan dibahas secara rinci di kelas, sisanya dipakai sebagai acuan dan bahan latihan membaca dan memahami program pada *web* dinamis. Karena dirancang sebagai buku kerja mahasiswa, mungkin diktat ini sulit untuk dipakai sebagai pegangan bagi pengajar. Rencananya, buku ajar ini akan dilengkapi dengan buku *pedagogi* yang ditujukan khusus sebagai pedoman bagi pengajar.

Notasi yang dipakai adalah notasi fungsional. Karena notasi fungsional tidak mempunyai eksekutor, mahasiswa harus mempunyai alat eksekusi dalam bahasa riil dan disadari bahwa akan sulit mendapatkan teks yang bebas kesalahan. Untuk latihan eksekusi, diktat ini dilengkapi dengan diktat lain yang berisi pedoman penerjemahan ke salah satu bahasa fungsional yang dipilih. Penulis menyadari bahwa diktat ini tentu masih ada banyak kesalahan karena ditambahkan program-program baru. Penulis mengharap agar kesalahan dapat disampaikan agar dapat diperbaiki pada versi berikutnya.

# PRAKATA

Alhamdulillah, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT karena buku ini telah selesai disusun. Buku ini disusun agar dapat membantu para mahasiswa dalam mempelajari konsep dan praktik pemrograman *web* dinamis, beserta mempermudah mempelajari materi pemrograman *web* dinamis terutama bagi kaum awam yang belum mengenal *web* dinamis itu sendiri.

Penulis pun menyadari jika di dalam penyusunan buku ini mempunyai kekurangan, namun penulis meyakini sepenuhnya bahwa sekecil apa pun buku ini tetap akan memberikan sebuah manfaat bagi pembaca.

Akhir kata untuk penyempurnaan buku ini, maka kritik dan saran dari pembaca sangat berguna untuk penulis kedepannya.

Magelang, 10 Desember 2020

Penulis

# DAFTAR ISI

KATA PENGANTAR.....	ii
PRAKATA .....	iv
DAFTAR ISI.....	v
BAB 1 PENGENALAN <i>HTML</i> .....	1
A. Tujuan Pembelajaran.....	1
B. <i>Web</i> , <i>Situs Web</i> , dan <i>Web Browser</i> .....	1
C. <i>WebServer</i> .....	2
D. <i>HTML</i> .....	2
E. Elemen dan <i>TagHTML</i> .....	3
F. <i>Tag-tag</i> Dasar <i>HTML</i> .....	4
G. <i>Hyperlink</i> .....	9
H. Tabel.....	9
I. Soal Latihan (Essay) .....	12
J. Soal Latihan (Pilihan Ganda) .....	12
BAB 2 KONSEP <i>PHP</i> .....	15
A. Tujuan Pembelajaran.....	15
B. Pengantar <i>PHP</i> .....	15
C. Aplikasi yang Diperlukan .....	16
D. Menjalankan <i>PHP</i> .....	17
E. Penulisan Coding <i>PHP</i> .....	17
F. Variabel.....	18
G. Konstanta .....	19
H. Komentar.....	19
I. Soal Latihan (Essay) .....	20
J. Soal Latihan (Pilihan Ganda) .....	21
BAB 3 TIPE DATA DAN OPERATOR.....	23
A. Tujuan Pembelajaran.....	23
B. Tipe Data di <i>PHP</i> .....	23
C. Operator di <i>PHP</i> .....	23
D. Soal Latihan (Essay) .....	27
E. Soal Latihan (Pilihan Ganda) .....	28

BAB 4 PERCABANGAN.....	31
A. Tujuan Pembelajaran .....	31
B. Struktur Percabangan.....	31
C. Struktur IF.....	31
D. Struktur Logika IF - ELSE .....	33
E. Struktur Logika ELSEIF .....	34
F. Struktur Logika <i>Switch</i> .....	35
G. Soal Latihan (Essay) .....	36
H. Soal Latihan (Pilihan Ganda) .....	37
BAB 5 PERULANGAN.....	39
A. Tujuan Pembelajaran .....	39
B. Struktur Perulangan .....	39
C. Struktur Perulangan For.....	39
D. Struktur Perulangan <i>While</i> .....	40
E. Struktur Perulangan Do - <i>While</i> .....	41
F. <i>Break</i> Dalam Perulangan .....	42
G. <i>Continue</i> Dalam Perulangan .....	42
H. Soal Latihan (Essay) .....	42
I. Soal Latihan (Pilihan Ganda) .....	43
BAB 6 ARRAY.....	47
A. Tujuan Pembelajaran .....	47
B. <i>Array</i> .....	47
C. Pendeklarasian <i>Array</i> .....	47
D. <i>IndexedArray</i> tanpa Key .....	48
E. <i>IndexedArray</i> dengan Key.....	48
F. <i>Associative Array</i> .....	48
G. <i>Array</i> Multidimensi .....	48
H. Menampilkan <i>ValueArray</i> .....	49
I. Menambahkan Elemen <i>Array</i> .....	49
J. Menghapus Elemen <i>Array</i> .....	50
K. Soal Latihan (Essay) .....	51
L. Soal Latihan (Pilihan Ganda) .....	51
BAB 7 FUNGSI.....	53
A. Tujuan Pembelajaran .....	53

B.	Fungsi.....	53
C.	Fungsi Dengan Argumen .....	53
D.	Fungsi Built-in.....	54
E.	Fungsi <i>String</i> .....	55
F.	Fungsi <i>Date</i> .....	57
G.	Penggunaan <i>Timestamp</i> .....	58
H.	Fungsi <i>Time</i> .....	59
I.	Fungsi <i>Mktime</i> .....	60
J.	Fungsi <i>Strtotime</i> .....	61
K.	Soal Latihan (Essay) .....	62
L.	Soal Latihan (Pilihan Ganda) .....	62
<b>BAB 8 FORM DAN HYPERLINK.....</b>		<b>65</b>
A.	Tujuan Pembelajaran.....	65
B.	Form .....	65
C.	<i>Text Field</i> .....	66
D.	<i>Password Field</i> .....	66
E.	Radio Button .....	66
F.	Check Box.....	67
G.	Submit Button .....	68
H.	Reset Button.....	68
I.	Select <i>List</i> (Drop-down <i>List</i> ) .....	69
J.	<i>Textarea</i> .....	69
K.	Metode <i>GET</i> dan <i>POST</i> .....	70
L.	Soal Latihan (Essay) .....	71
M.	Soal Latihan (Pilihan Ganda) .....	72
<b>BAB 9 KONEKSI DATABASE.....</b>		<b>73</b>
A.	Tujuan Pembelajaran.....	73
B.	<i>MySQL</i> .....	73
C.	Tipe Data <i>MySQL</i> .....	74
D.	Koneksi <i>PHP</i> dan <i>MySQL</i> .....	75
E.	<i>PHP Data Objects</i> .....	77
F.	Langkah-langkah Koneksi <i>MySQL</i> .....	77
G.	Session.....	82
H.	Cookies .....	84
I.	Soal Latihan (Essay) .....	86

J.	Soal Latihan (Pilihan Ganda) .....	86
BAB 10	KONSEP OOP .....	89
A.	Tujuan Pembelajaran .....	89
B.	OOP.....	89
C.	Perbedaan OOP dan Prosedural .....	90
D.	Class, <i>Object</i> , Property, dan Method .....	90
E.	Variabel \$this.....	92
F.	Enkapsulasi.....	92
G.	Inheritance (pewarisan atau penurunan).....	92
H.	Constructor.....	92
I.	Destructor .....	93
J.	Soal Latihan (Essay) .....	93
K.	Soal Latihan (Pilihan Ganda) .....	93
BAB 11	CSS.....	95
A.	Tujuan Pembelajaran .....	95
B.	CSS .....	95
C.	Metode Penerapan CSS.....	96
D.	Soal Latihan (Essay) .....	97
E.	Soal Latihan (Pilihan Ganda) .....	97
BAB 12	Javascript .....	99
A.	Tujuan Pembelajaran .....	99
B.	Javascript.....	99
C.	jQuery .....	100
D.	Soal Latihan (Essay) .....	101
E.	Soal Latihan (Pilihan Ganda) .....	101
DAFTAR PUSTAKA	.....	102
GLOSARIUM	.....	105
INDEKS	.....	109
BIOGRAFI PENULIS	.....	115

# BAB 1

## PENGENALAN HTML

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa konsep umum *HyperText Markup Language (HTML)*.
2. Mahasiswa dapat mengetahui dan memahami bahasa *HTML* sebagai dasar pemrograman untuk membuat *web* statis.
3. Mahasiswa dapat memulai membuat *web* dengan bahasa *HTML*.

### B. Web, Situs Web, dan WebBrowser

*Web* merupakan kumpulan dokumen multimedia yang berupa teks, gambar, suara, animasi, video, yang di dalamnya menggunakan protokol *Hypertext Transfer Protocol (HTTP)*. *World Wide Web* termasuk salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet.

Pengguna komputer dapat berinteraksi dengan pengguna internet lainnya dan mencari informasi yang diperlukan menggunakan situs *web* yang merupakan kumpulan dari dokumen *web* yang telah dipublikasi ke internet atau intranet, dan telah memiliki *Unified Resource Locator (URL)* yang dapat diakses oleh setiap orang pengguna internet dengan cara mengetikkan alamat *URL*nya pada *webbrowser*.

*WebBrowser* yaitu *software* yang mampu menjalankan dokumen-dokumen *web* yang memiliki *format HTML*, prosesnya yaitu dilakukan oleh komponen *webengine* yang terdapat pada aplikasi *browser*. Cara kerja *webbrowser* dengan menerjemahkan kumpulan dokumen *web* yang ditampilkan oleh *browser*. Contoh dari *webbrowser*: Opera, Mozilla, Netscape Navigator, Internet Explorer, Lynx, Konqueror, dan Safari.

## C. WebServer

*WebServer* adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui *browser* kemudian memberikan tanggapan permintaan dalam bentuk halaman situs *web* atau lebih umumnya dalam dokumen *HTML*. Namun, *webserver* dapat mempunyai dua pengertian berbeda, yaitu sebagai bagian dari perangkat keras (*hardware*) maupun sebagai bagian dari perangkat lunak (*software*). *Webserver* digunakan untuk menyimpan semua data seperti *HTML* dokumen, gambar, *file CSSstylesheets*, dan *file JavaScript*. Sedangkan pada sisi *software*, fungsi *webserver* adalah sebagai pusat kontrol untuk memproses permintaan yang diterima dari *browser*. Komputer ini akan melayani permintaan dokumen *web* dari kliennya. Contoh dari *webserver*: ISS, PWS, Apache, Xitami.

## D. HTML

*Hypertext Markup Language (HTML)* adalah salah satu yang memerintahkan bahasa pemrograman *web* desain juga biasa disebut *script* untuk menyusun dokumen-dokumen *web*. Dokumen ini dikenal dengan *webpage*, selanjutnya dokumen *HTML* disajikan pada *webbrowser*. Ada dua cara untuk membuat sebuah *webpage*:

1. Menggunakan editor teks biasa (notepad, editplus, sublime)
2. Menggunakan *HTML* Editor.

Dokumen *HTML* disusun oleh elemen-elemen seperti: *head*, *body*, tabel, paragraf dan *list*. Untuk menandai berbagai elemen dalam suatu dokumen *HTML*, kita menggunakan *tag*. *Tag* merupakan kode yang digunakan untuk *markup* teks *ASCII* menjadi *file HTML*. *TagHTML* terdiri atas sebuah kurung sudut kiri (<, lebih kecil), sebuah nama *tag*, dan sebuah kurung sudut kanan (>, tanda lebih besar).

*Tag* pembuka yaitu <*html*> dan ada *tag* penutup yaitu </*html*> yang ditandai dengan tanda *slash* (garis miring) di awal tulisan. *Tag* di atas memberikan aturan bahwa yang ditulis di antara kedua *tag* tersebut adalah isi dari dokumen *HTML*. Dokumen *HTML* yang ditampilkan pada *webbrowser* dibuat dengan nama ekstensi (*.html*) atau (*.htm*)

## E. Elemen dan TagHTML

Dokumen *HTML* terdiri dari beberapa elemen, yang dapat berupa teks murni, atau bukan teks, atau keduanya, namun juga dapat berisi *tag* lain. Elemen yang dibutuhkan untuk membuat suatu dokumen *HTML* dinyatakan dengan *tag* `<html>`, `<head>`, dan `<body>` berikut *tag-tag* pasangannya. Setiap dokumen terdiri atas *taghead* dan *body*. Elemen *head* berisi informasi tentang dokumen tersebut, dan elemen *body* berisi teks yang sebenarnya yang tersusun dari *link*, grafis, paragraf dan elemen lainnya.

*Tag* umumnya berpasangan, misalkan *tag* awal `<H1>` berpasangan dengan *tag* akhir `</H1>`, tanda slash (garis miring) memberikan tanda bahwa *tag* tersebut merupakan pembatas akhir elemen yang dibuka oleh *tag* awal. Jadi, *tag* ditulis dengan `<nama tag> ... </nama tag>` selain itu dalam penamaan *tag* tidak *casesensitive*.

Berikut ini adalah pola dasar sebuah dokumen *HTML*:

```
<html>
<head>
    ...Informasi tentang dokumenHTML.
</head>
<body>
    ...Informasi yang akan ditampilkan dalam
    webbrowser.
</body>
</html>
```

Penulisan elemen yang diawali dan diakhiri dengan tanda *tagHTML*, memiliki syarat penulisan yaitu :

1. *TagHTML* diapit dengan dua karakter kurung sudut (`<` dan `>`)
2. *TagHTML* secara normal selalu berpasangan, misal (`<i>...</i>`)
3. *TagHTML* tidak *casesensitive*, misal `<b>` sama dengan `<B>`
4. Dalam suatu *tag* terdapat *tag* lagi, maka penulisan *tag* akhir tidak boleh bersilang, harus berurutan. Misalnya `<b><l>...  
Tebal dan Miring... </l></b>`

Di dalam *tag* terdapat atribut, dan atribut merupakan informasi tambahan yang diberikan kepada *tag*. Informasi ini bisa berupa instruksi untuk warna dari *text*, besar huruf dari *text*. Atribut digunakan

untuk mengubah *default* pemformatan dengan *tag* yang bersangkutan. Setiap atribut memiliki pasangan nama dan nilai (*value*), dan ditulis dengan `name="value"`. *Value* diapit tanda kutip, boleh tanda kutip satu (') atau dua (").

## F. Tag-tag Dasar HTML

*HTML* memiliki *tag-tag* dasar yang dapat digunakan sebagai acuan dalam memulai menggunakan *HTML*.

### 1. *HTML*

Merupakan *tag* awal yang mendefinisikan bahwa dokumen adalah dokumen *html*. Penulisan *tag* seperti berikut ini :

```
<html>
...
</html>
```

### 2. *Head*

Merupakan *tag* berikutnya setelah *taghtml*, digunakan untuk menuliskan keterangan tentang dokumen *web* yang akan ditampilkan. Penulisan *tag* seperti berikut ini :

```
<html>
<head>
...
</head>
...
</html>
```

### 3. *Title*

*Title* dituliskan di dalam *head* pada dokumen *HTML*, yang akan muncul pada *caption* halaman *browser* jika halaman tersebut di akses. Penulisan *tag* seperti berikut ini :

```
<html>
<head>
<title> Halaman Judul Dokumen</title>
</head>
...
</html>
```

#### 4. *Body*

*Body* merupakan section utama dalam dokumen *web* yang akan ditampilkan di dalam *browser*. Penulisan *tag* seperti berikut ini :

```
<html>
<head>
<title> Halaman Judul Dokumen</title>
</head>
<body>
    ... isi dokumen
</body>
</html>
```

#### 5. Heading

*Tagheading* (hx) digunakan untuk membuat hormat *heading* judul dan sub judul pada suatu halaman *web*. Pada *HTML* terdapat enam buah, yaitu dari (h1) sampai dengan (h6)

```
<html>
<head>
<title> Halaman Judul Dokumen</title>
</head>
<body>
    <h1> Heading 1 </h1>
    <h2> Heading 2 </h2>
    <h3> Heading 3 </h3>
    <h4> Heading 4 </h4>
    <h5> Heading 5 </h5>
    <h6> Heading 6 </h6>
</body>
</html>
```

#### 6. Paragraf

*Tag* paragraf <p> digunakan untuk melakukan pengaturan antar paragraf yang ada pada halaman *web*. Dalam elemen paragraf terdapat atribut : *align*=[*left* | *center* | *right*] yang berfungsi sebagai pengatur perataan paragraf. Pada elemen paragraf dapat ditambahkan *tag* horizontal ruler<hr>, untuk menampilkan garis horizontal tiga dimensi di dalam halaman *web*. *Tag*<hr> tidak memerlukan elemen penutup.

```
<html>
<head>
<title> Paragraf dan Horizontal Ruler </title>
</head>
<bodycolor = "#222222" text = "#FF0000">
<h1> Modul Web Dinamis dengan PHPMySQL</h1>
```

```
<hr width = "80%" align = "left">
<p>Web dinamis adalah jenis web yang content/isinya
dapat berubah-ubah setiap saat </p>
</body>
</html>
```

Dalam paragraf di halaman *web*, terdapat *tagBreak*`<br>` yang digunakan untuk memberikan baris baru. Pada *tagbreak* tidak memerlukan *tag* penutup *break*.

```
<html>
<head>
<title>Break</title>
</head>
<body bgcolor = "#222222" text = "#FF0000">
<h1> Modul Web Dinamis dengan PHPMySQL</h1>
<hr width = "80%" align = "left">
<p>Web dinamis adalah jenis web yang content/isinya
dapat berubah-ubah setiap saat. <br>Web dinamis dapat
menampilkan animasi flash </p>
</body>
</html>
```

## 7. List

List digunakan dalam halaman *HTML* untuk membuat daftar atau mengelompokkan data. Dalam *taglist* terdapat 2 jenis, yaitu:

### a) *OrderedList* (Numbering)

*TagOrderedList* diawali dengan `<ol>` dan diakhiri dengan `</ol>` digunakan untuk membuat daftar dimana setiap bagiannya menggunakan nomor secara terurut, dimulai dengan *tag*`<ol>` dan diakhiri dengan *tag*`</ol>`. Untuk menyatakan tiap bagiannya, digunakan *tag*`<li>` dan tidak diperlukan *tag* penutup. Atribut *elementlist* yaitu :

*TYPE* = [ 1 | a | A | i | I ] default 1

```
<html>
<head>
<title>OrderedList</title>
</head>
<body>
<h3> Nama Program Studi </h3>
<ol type="A">
<li> Teknik Informatika
<li> Teknik Industri
<li> Teknik Mesin
<li> Teknik Elektro
```

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

b) *UnorderedList* (Bullet)

*TagUnorderedList* diawali dengan `<ul>` dan diakhiri dengan `tag</ul>` digunakan untuk membuat daftar dengan tanda bullet, berbeda dengan *orderedlist* karena tidak terlihat daftar nomor urutan. Untuk menyatakan tiap bagiannya, digunakan `tag<li>` dan tidak diperlukan `tag` penutup. Atribut *elementlist* yaitu :

*TYPE* = [ *disc* | *square* | *circle* ] *defaultdisc*

```
<html>
<head>
<title>OrderedList</title>
</head>
<body>
<h3> Nama Program Studi </h3>
<ul type="square">
<li> Teknik Informatika
<li> Teknik Industri
<li> Teknik Elektro</ul>
</body>
</html>
```

8. Format Karakter

Format karakter pada halaman *HTML* yang ditampilkan akan sangat berguna untuk membuat sebuah *web* lebih menarik. Pemformatan karakter tersebut mencakup beberapa hal, yaitu :

a) *LogicalFormat*

*Logicalformat* digunakan untuk menerapkan layout secara logis dan terstruktur. *Tag-tag* yang termasuk dalam *logicalformat* adalah sebagai berikut :

<b>TagHTML</b>	<b>Keterangan</b>
<code>&lt;cite&gt;</code>	Menandai suatu kutipan (sitasi)
<code>&lt;code&gt;</code>	Menampilkan kode-kode pemrograman termasuk bahasa <i>PHP</i>
<code>&lt;em&gt;</code>	Menandai suatu teks yang ditekankan oleh penulis
<code>&lt;kbd&gt;</code>	Menandai suatu teks yang harus dimasukkan oleh user melalui <i>keyboard</i>
<code>&lt;samp&gt;</code>	Menandai suatu teks yang digunakan sebagai contoh

<strong>	Menandai bagian yang terpenting dari suatu teks
<var>	Menampilkan nama variabel
<dfn>	Menandai sebuah subdefinisi dari daftar definisi

b) *PhysicalFormat*

*Physicalformat* digunakan untuk memformat suatu font. *Tag-tag* termasuk dalam *physicalformat* adalah sebagai berikut :

<b>TagHTML</b>	<b>Keterangan</b>
<b>	Menampilkan huruf tebal
<i>	Menampilkan huruf miring
<u>	Menampilkan garis bawah pada teks
<tt>	Menampilkan huruf seperti huruf mesin ketik
<strike>	Menampilkan garis horizontal pada bagian tengah huruf
<big>	Menampilkan ukuran huruf yang lebih besar
<small>	Menampilkan ukuran huruf yang lebih kecil
<sub>	Menampilkan subscript
<sup>	Menampilkan superscript

c) *Preformatted Text*

*Preformatted text*<pre> digunakan untuk menampilkan teks di *webbrowser* dalam font monospaced, yaitu seperti terlihat dalam teks yang ditampilkan oleh aplikasi berbasis teks.

9. Font

*Tag*<font> digunakan untuk menentukan *format* tampilan font dalam halaman *HTML* yaitu dengan atribut *color*, *size*, *style* dan lainnya.

a) Atribut *Color*, digunakan untuk mengatur warna font sesuai dengan keinginan. Untuk memberi nilai pada atribut *color* ada dua cara, yaitu :

- 1) Dengan menyebutkan jenis warna, seperti *red*, *green*, dan *blue*.
- 2) Dengan menggunakan nilai *RGB (Red, Green, Blue)* dari suatu warna, misalnya FF0000 untuk *red*, 00FF00 untuk *green*, dan 0000FF untuk *blue*.

b) Atribut *Size*, digunakan untuk mengatur ukuran font pada dokumen *HTML* yang ditampilkan pada *webbrowser*. Ukuran

font dimulai dari huruf terkecil yaitu 1 dan nilai 7 untuk ukuran font yang paling besar.

- c) Atribut *Face* digunakan untuk mengatur jenis huruf, seperti Arial, Times New Roman, Verdana, dan sebagainya.

```
<html>
<head>
<title>Format Font </title>
</head>
<body>
  <p><font color="#FF0066" size="5"
  Mengatur warna font dan ukuran
  font</font></p>
  <p><font face="Arial" size="3"
  Mengatur jenis huruf dan ukuran
  font</font></p>
</body>
</html>
```

## G. Hyperlink

*Hyperlink* digunakan untuk menghubungkan beberapa halaman *web*. Anchor (a) berfungsi sebagai awal penulisan *hyperlink*. Dalam sebuah halaman dokumen *HTML*, dapat dibuat *link* yaitu dengan nama anchor lokasi tujuan dari *link* tersebut.

1. Untuk membuat *link* ke halaman dokumen *HTML* lain digunakan perintah berikut ini :

```
<a href="nama_dokumen"> Halaman teks pada browser</a>
```

2. Untuk membuat *link* dari dokumen *HTML* untuk memuat banyak informasi dengan perintah berikut ini :

```
<a name ="HTML dan CSS">HTML dan CSS</a>
```

Cara melakukan *link* ke bagian tersebut :

```
<a href="nama_anchor"> Halaman teks pada browser</a>
```

## H. Tabel

Tabel banyak digunakan untuk menampilkan sebuah informasi yang lebih ringkas dan mudah dibaca. Untuk membuat tabel digunakan *tag* awal `<table>` dan *tag* akhir `</table>`. *Tag*`<table>` memiliki beberapa bagian penting, yaitu:

1. `<caption> ... </caption>` digunakan untuk membentuk judul tabel. Judul tabel ini akan terletak di luar tabel, bisa dibagian atas atau di bagian bawah.
2. `<th> ... </th>` digunakan untuk meletakkan judul tabel di bagian paling atas atau paling kiri dari suatu tabel. Tabel *header* akan dicetak dalam huruf tebal.
3. `<tr> ... </tr>` dipakai untuk membentuk garis pada suatu tabel.
4. `<td> ... </td>` digunakan sebagai tempat menulis data atau informasi dalam tabel.

<b>Atributtag&lt;table&gt;</b>	<b>Keterangan</b>
<code>align</code>	Posisi horizontal tabel
<code>Background</code>	Menentukan gambar latar belakang tabel
<code>Bgcolor</code>	Menentukan warna latar belakang tabel
<code>Border</code>	Menentukan tebal bingkai tabel
<code>Bordercolor</code>	Menentukan warna bingkai tabel
<code>bordercolorlight</code>	Menentukan warna depan bingkai tabel
<code>bordercolordark</code>	Menentukan warna bayangan bingkai tabel
<code>cellspacing</code>	Menentukan jarak spasi antar sel
<code>cellpadding</code>	Menentukan jarak isi sel dengan bingkai
<code>height</code>	Menentukan tinggi tabel
<code>widht</code>	Menentukan lebar tabel
<code>valign</code>	Menentukan posisi vertikal teks dalam sel

<b>Atributtag&lt;td&gt;</b>	<b>Keterangan</b>
<code>align</code>	Posisi horizontal sel
<code>Background</code>	Menentukan gambar latar belakang sel
<code>Bgcolor</code>	Menentukan warna latar belakang sel
<code>Border</code>	Menentukan tebal bingkai sel
<code>Bordercolor</code>	Menentukan warna bingkai sel
<code>bordercolorlight</code>	Menentukan warna depan bingkai sel
<code>bordercolordark</code>	Menentukan warna bayangan bingkai sel
<code>colspan</code>	Menentukan jumlah kolom yang digabung
<code>height</code>	Menentukan tinggi sel
<code>nowspan</code>	Menentukan teks agar tetap satu baris
<code>rowspan</code>	Menentukan lebar tabel
<code>valign</code>	Menentukan posisi vertikal teks dalam sel

## Contoh *script* penggunaan Tabel :

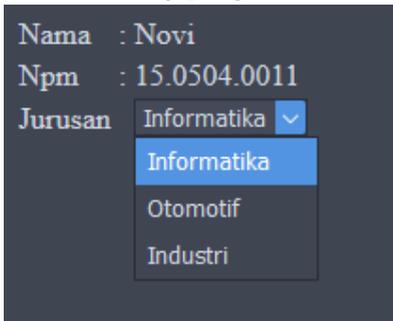
```
<html>
<head>
<title>Format Font </title>
</head>
<body>Tabel Colspan
<table border="1" cellspacing="2" width="50%">
<tr><td colspan="3" align="center" BGCOLOR="yellow">
Quarter 1</td></tr>

<tr><td>Januari</td><td>Februari</td>
<td BGCOLOR="skyblue"> Maret</td></tr>
</table>

<br>
Tabel Rowspan
<table border="1" cellspacing="3" width="40%">
<tr BGCOLOR="yellow"><td rowspan="3">Quarter-
1</td><td>Januari</td></tr>
<tr><td align="center">Februari</td></tr>
<tr><td align="left"
BGCOLOR="skyblue">Maret</td></tr></table>
</body>
</html>
```

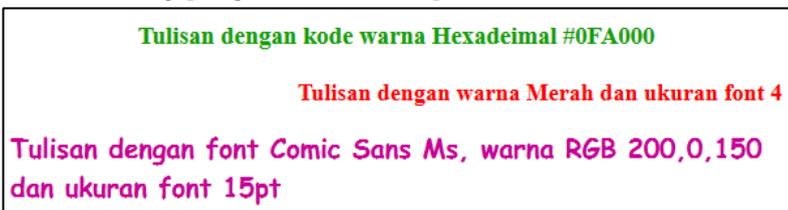
## I. Soal Latihan (Essay)

- 1) Buatlah *listing* program untuk tampilan di bawah ini :



Nama : Novi  
Npm : 15.0504.0011  
Jurusan : Informatika  
Informatika  
Otomotif  
Industri

- 2) Buatlah *listing* program untuk tampilan di bawah ini :



Tulisan dengan kode warna Hexadeimal #0FA000  
Tulisan dengan warna Merah dan ukuran font 4  
Tulisan dengan font Comic Sans Ms, warna RGB 200,0,150 dan ukuran font 15pt

- 3) Jelaskan perbedaan *tagHTML* echo, print dan printf. Buktikan perbedaan tersebut dengan script yang sesuai!

## J. Soal Latihan (Pilihan Ganda)

- 1) Berikut ini yang bukan merupakan bagian dari *taglisting* pada *HTML* adalah?
- <ul>
  - <al>
  - <li>
  - <dl>
- 2) *Tag* yang berfungsi untuk membuat judul pada tabel adalah...
- <Caption>
  - <Title>
  - <Tr>
  - <Head>

- 3) Berikut ini merupakan tipe data yang dapat digunakan untuk penulisan nama adalah?
- Float
  - Text*
  - Double*
  - Integer*
- 4) Apakah fungsi *cellpadding*?
- Mengatur jarak dari border sisi dalam tabel dengan isi *text* table
  - Mengatur lebar kolom dalam tabel
  - Mengatur jarak dari border sisi luar dengan *border* sisi dalam
  - Mengatur jarak setiap kolom dalam tabel
- 5) Berikut yang **bukan** merupakan syarat penulisan *tagHTML* adalah?
- TagHTMLcasesensitive*, misalnya penggunaan <B>
  - TagHTML* diapit dengan dua karakter kurung sudut (< dan >)
  - TagHTML* secara normal selalu berpasangan, misal (<i>...</i>)
  - Penulisan *tag* awal dan akhir tidak boleh bersilang, dan harus berurutan.



# BAB 2

## KONSEP PHP

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa bahasa pemrograman *Web* yaitu *PHP*.
2. Mahasiswa dapat mengetahui sintaks penulisan bahasa *PHP*, penggunaan variabel, dan memasukkan komentar pada script *PHP*.
3. Mahasiswa dapat memulai membuat *web* dengan bahasa *PHP*.

### B. Pengantar *PHP*

*PHP* merupakan akronim dari "*PHP* : Hypertext Presprocessor". *PHP* adalah bahasa pemrograman *web* bersifat *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* dinamis. Artinya bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (*WebServer*) sedang hasilnya yang dikirimkan ke komputer *client* (*WebBrowser*) dalam bentuk *scriptHTML* (*Hypertext Markup Language*) tanpa mengetahui kode asli. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh user sehingga keamanan *web* lebih terjamin, dan dengan *web* dinamis ini halaman *web* dapat menampilkan isi basis data yang telah tersimpan. Selain *PHP*, ada beberapa bahasa *server-side scripting*, diantaranya : *ASP.NET*, *JSP*, *JSF*, *Perl*, *Python*, *Ruby on Rail*, *Node.js*, *Zope*.

*PHP* termasuk dalam *open source*, sehingga *source code PHP* dapat diubah dan didistribusikan secara bebas. Versi terbaru *PHP* dapat di unduh gratis di situs resmi *PHP* : <http://www.php.net>. Awalnya, *PHP* bekerja di lingkungan *Linux*, karena merupakan 'turunan' bahasa *PEARL* yang merupakan bahasa pemrograman *native* di lingkungan *Linux* dan *Unix*. Namun seiring perkembangan dan kebutuhan maka *PHP* dibuat

secara *cross platform*, baik Linux, Windows, Macintosh. *PHP* dapat dibangun sebagai modul pada *webserver* Apache dan sebagai *binary* yang dapat berjalan sebagai Common Gateway Interface(CGI). *PHP* dapat mengirim *HTTP Header*, dapat mengatur *cookies*, mengatur *authentication* dan *redirect users*.

*PHP* memiliki keunggulan yaitu dapat melakukan konektifitas yang baik dengan beberapa *Database Management System (DBMS)*, antara lain dengan Oracle, *MySQL*, Microsoft *SQL Server*, PostgreSQL, dBASE, dan tak terkecuali semua *database* berinterface ODBC.

### C. Aplikasi yang Diperlukan

Untuk dapat bekerja dengan *PHP*, berikut ini adalah beberapa aplikasi yang perlu disiapkan/diinstall di dalam komputer, antara lain :

1. *Browser* dibutuhkan untuk menjalankan dan menampilkan contoh kode program yang dibuat. Berikut beberapa aplikasi *webbrowser* yang sering digunakan : Mozilla Firefox, Google Chrome, Safari, Opera, Internet Explore.
2. *Webserver*. *WebServer* merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dari klien dan mengirimkan kembali dalam bentuk dokumen *HTML*. Beberapa *webserver* : Apache, Nginx, IIS, LiteSpeed, Tomcat, Lighttpd. Tool aplikasi yang di dalamnya sudah terdapat *webserver* (Apache), *PHP*, dan *MySQL* yang terintegrasi menjadi satu, yaitu : *XAMPP*, *WAMP*, *LAMP*, *MAMP*.
3. *Database server*. Salah satu kelebihan dari *PHP* adalah mampu berkomunikasi dengan berbagai *database* yang terkenal, sehingga *web* bersifat dinamis. Database tersebut adalah : *MySQL*, Ms.*SQL*, Oracle, Postgres, Sybase, Velocis, dBase, DBM, FilePro.
4. *Web Editor* merupakan aplikasi yang digunakan untuk mengetikkan source code *PHP*, *HTML* dan Javascript. Saat ini banyak aplikasi yang dikembangkan sebagai *web* editor, seperti : Notepad, Notepad++, SublimeText, Adobe Dreamweaver.

## D. Menjalankan *PHP*

Sebelum memulai untuk menggunakan *PHP*, langkah awal yang harus disiapkan, yaitu :

1. Melakukan *install* aplikasi *XAMPP*, aktifkan *webserver* (Apache) dan database (MySQL).
2. Selanjutnya membuat folder penyimpanan *file-file PHP*, untuk memudahkan melakukan *compilewebsite*.
3. Konfigurasi program adalah berada pada C: \Program Files\XAMPP
4. Kode *PHP* diletakan pada folder C:\Program Files\xampp\htdocs

## E. Penulisan Coding *PHP*

Script *PHP* termasuk dalam *HTML-embedded*, yaitu kode *PHP* dapat disisipkan ke dalam halaman *HTML*. Ada beberapa cara untuk menuliskan script *PHP*, yaitu:

1. Harus ditulis di antara *tagPHP*, berikut *tag* yang bisa digunakan:

```
1. <?php
    script PHP ?>
2. <script language="php"> .... </script>
3. <% script PHP %>
4. <? script PHP ?>
```

Cara penulisan *script* yang pertama, merupakan *format* yang dianjurkan dalam penulisan php. Cara yang kedua lebih sering digunakan karena lebih ringkas dalam penulisan. Cara yang ketiga digunakan untuk mengantisipasi editor-editor yang tidak dapat menerima kedua cara diatas. Selain itu kita juga dapat menggunakan cara penulisan ASP.

2. Setiap satu *statement* (perintah) biasanya diakhiri dengan titik-koma (;)
3. *Case Sensitive* untuk nama *identifier* yang dibuat oleh *user* (berupa variable, konstanta, fungsi dll), namun **TIDAK Case Sensitive** untuk *identifier* built-in dari *PHP*. Jadi :

```
echo = ECHO
while = WHILE
```

```
$nama ≠ $Nama ≠ $NAMA  
hitungLuas() ≠ HitungLuas()
```

```
<html>  
<head>  
<title>Coba Penyisipan PHP Pada HTML</title>  
</head>  
<body>  
    <?php  
        echo "Ini skrip PHP pertamaku!";  
    ?>  
</body>
```

## F. Variabel

Variabel adalah suatu pengenal dalam program yang berfungsi untuk menyimpan nilai secara sementara dan dapat berubah selama program dijalankan. Tidak seperti bahasa pemrograman lainnya yang mengharuskan kita untuk mendeklarasikan variabel terlebih dahulu, variabel dalam *PHP* tidak harus dideklarasikan sebelum variabel tersebut digunakan. Variabel diwakili oleh kata tertentu dengan aturan penulisan sebagai berikut :

1. Variabel diawali dengan simbol dollar (\$).
2. Sintak penulisan variabel : \$NamaPengenal = nilai.
3. Nama variabel harus diawali dengan huruf atau *underscore* (\_).
4. Karakter berikutnya bisa terdiri dari huruf, angka dan *underscore*.
5. Huruf kecil dan besar dibedakan (*casesensitive*), tidak boleh ada spasi.
6. Tidak boleh menggunakan kata yang merupakan fungsibuilt-in*PHP*.
7. Tidak boleh menggunakan tanda baca.

Untuk menampilkan data dalam *standard output* bisa menggunakan perintah echo. Fungsi *print* memiliki fungsi yang sama dengan echo, namun hanya mampu menampilkan data tunggal (hanya satu data).

Sintak-nya : echo(data); atau echo data; Fungsi echo digunakan untuk menampilkan *output* sebanyak satu atau lebih data yang dipisahkan dengan tanda koma (,). *Boolean printf (stringformat [, mixed args])*. Fungsi *printf* memiliki fungsi yang sama dengan echo dan print, namun fungsi ini dapat mengatur *format* data. Pada *printf*, parameter

pertama menunjukkan *formatstring*, dan parameter ke-2,3 dst... menunjukkan argumen atau data.

```
<html>
<head><title>echo dan print</title></head>
<body>
  <?php
    echo "Nama : " , "Putri Malu" , "<br>" ;
    print "NIM : " ;
    print "19.0504.1234" ;
  ?>
</body>
</html>
```

## G. Konstanta

Konstanta merupakan *variable* dengan nilai tetap. Konstanta diberikan nilai awal program dan nilainya tidak akan berubah selama program berjalan. *PHP* telah mendefinisikan beberapa konstanta, misalnya : *PHP\_version*, yaitu konstanta yang memberikan informasi tentang versi *PHP* yang digunakan. Selain konstanta yang telah dibuat, pengguna juga dapat membuat konstanta sendiri. Aturan penulisan konstanta adalah sebagai berikut :

```
define("nama_konstanta", nilai_konstanta")
```

Contoh script dengan konstanta :

```
<html>
<head><title>konstanta</title></head>
<body>
  <?php
    define ("NAMA", "Putri Malu");
    define ("NILAI", 90);
    echo "Nama : " . NAMA;
    echo "<br>Nilai : " . NILAI;
  ?>
</body>
</html>
```

## H. Komentar

Jika ada sebuah perintah dalam suatu kode *PHP* tidak akan dieksekusi, maka perintah tersebut dapat dijadikan sebuah komentar. Penggunaan komentar sangat dianjurkan untuk memberi penjelasan/keterangan pada kode-kode program *PHP* yang diketikkan

dalam sebuah dokumen *web*. Ada tiga cara penggunaan komentar, yaitu:

ScriptPHP	Keterangan
// komentar	Digunakan untuk komentar satu baris, dengan menggunakan tanda miring <i>double/slash</i>
# komentar	Digunakan untuk komentar satu baris, dengan menggunakan tanda kres
/* komentar */	Digunakan untuk komentar lebih dari satu baris, menggunakan tanda [ /* ] dan diakhiri dengan [ */ ]

```
<html>
<head>
<title>Coba Penyisipan PHP Pada HTML</title>
</head>
<body>
    <?php
        // contoh komentar satu baris
        # contoh lain komentar satu baris

        echo "Ini skrip PHP pertamaku!";

        /* ini adalah
        Contoh komentar
        Lebih dari satu baris */

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

## I. Soal Latihan (Essay)

1. Pada gambar di bawah terdapat 2 variable bertipe *String*, dan di cetak menggunakan *echo*. Buatlah ScriptPHP nya !

```
Nama saya Novi
Npm saya 15.0504.0011
```

2. Didalam pemrograman *PHP* terdapat *Variable* dan Konstanta yang mana *variable* merupakan pengenalan dalam program yang berfungsi untuk menyimpan nilai secara sementara dan dapat berubah sedangkan konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Tuliskan *script php* yang berisikan sebuah *variable* dan konstanta.
3. Buatlah *script PHP* dengan menambahkan Variabel dan Konstanta untuk tampilan di bawah ini :

Hasil perkalian 3.25 dan 2.555  
adalah 8.3037

## J. Soal Latihan (Pilihan Ganda)

1. Berikut ini merupakan script untuk menampilkan sebuah output pada bahasa pemrograman php yang berisikan lebih dari satu data, manakah script yang dapat berjalan dan tidak terjadi eror?

a.

```
<?php
    echo "Data1","Data2","Data3";
?>
```

b.

```
<?php
    echo ("Data1","Data2","Data3");
?>
```

c.

```
<?php
    print ("Data1","Data2","Data3");
?>
```

d.

```
<?php
    print "Data1","Data2","Data3";
?>
```

2. Tanda \$ pada php di gunakan untuk...

- Komentar
- Konstanta
- Koneksi Database
- Variabel

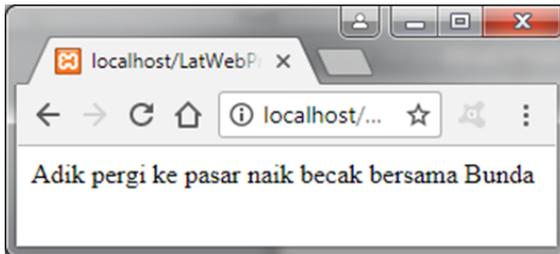
3. Manakah script yg sesuai untuk melengkapi script *PHP* diatas agar menghasilkan tampilan seperti berikut!

```
<?php
    $a = "pasar";
    $b = "Bunda";
    $c = "becak";
    $stampil = ".....";
    echo $stampil;
?>
```

- ```
$stampil = "Adik pergi ke ".$a."naik ".$c."bersama ".$b;
```
- ```
$stampil = "Adik pergi ke "+$a+" naik "+$c+" bersama "+$b;
```

- c. `$stampil = "Adik pergi ke $a naik ".$c." bersama ".$b;`
- d. `$stampil = "Adik pergi ke $a naik $b bersama $c";`

4. *Script* php untuk tampilan di bawah ini adalah:



- a. `$stampil = "Adik pergi ke".$a."naik".$c."bersama".$b;`
- b. `$stampil = "Adik pergi ke "+$a+" naik "+$c+" bersama "+$b;`
- c. `$stampil = "Adik pergi ke $a naik ".$c." bersama ".$b;`
- d. `$stampil = "Adik pergi ke $a naik $b bersama $c";`

5. Berikut yang bukan atribut *type* yang terdapat dalam *tag* input adalah?

- a. `<input type = 'text'>`
- b. `<input type='password'>`
- c. `<input type='button'>`
- d. `<input type='textarea'>`

# BAB 3

## TIPE DATA DAN OPERATOR

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa tipe data dan operator pada *PHP*.
2. Mahasiswa dapat mengetahui sintaks penulisan bahasa *PHP*, penggunaan tipe data, dan operator pada script *PHP*.

### B. Tipe Data di *PHP*

*PHP* mengenal beberapa macam tipe data antara lain *Integer*, *FloatingPoint*, dan *String*. *Floatingpoint* lebih dikenal dengan nama *double* atau desimal. Penulisan *string* selalu diawali dengan tanda petik ganda ( " ) atau tanda petik tunggal ( ' ).

Contoh-contoh penulisan tipe data, yaitu sebagai berikut :

Tipe Data	Contoh	Keterangan
<i>Integer</i>	<code>\$sks = 24;</code> <code>\$nilai = 90;</code>	Jenis data bilangan bulat, maksimum 32 bit
<i>Double</i>	<code>\$ipk = 3.88;</code> <code>\$pajak = 20.00;</code>	Jenis data pecahan dan desimal
<i>String</i>	<code>\$jurusan = "Teknik Informatika"</code> <code>\$jenjang = "Strata Satu"</code>	Jenis data teks / untaian karakter, kalimat
<i>Boolean</i>	<code>\$benar = true;</code> <code>\$salah = false;</code>	Bilangan dengan hasil <i>TRUE</i> dan <i>FALSE</i>

### C. Operator di *PHP*

Operator adalah suatu simbol yang berfungsi untuk memanipulasi data, seperti penambahan, pengurangan, perkalian, perbandingan, atau penugasan. Sedangkan yang dioperasikan operator disebut dengan

*operand*. Adapun macam – macam operator yaitu :

1. Operator Aritmatika adalah operator yang berhubungan dengan fungsi matematika. Operator ini sering digunakan dalam program yang akan dibuat. Pada tabel dibawah ini, akan diberikan contoh penggunaan operator aritmatika. Misalnya  $y = 6$ .

Operator	Fungsi	Contoh	Output
+	Penjumlahan	$x = y + 4$	$x = 10$
-	Pengurangan	$x = y - 4$	$x = 2$
*	Perkalian	$x = y * 4$	$x = 24$
/	Pembagian	$x = y / 2$	$x = 3$
%	Modulus / Sisa Pembagian	$x = y \% 2$	$x = 0$
++	Increment / Peningkatan	$x = y ++$	$x = 7$
--	Decrement / Penurunan	$x = y --$	$x = 5$

Berikut ini tabel *precedence of arithmetic operator* di *PHP*.

Prioritas	Operator	Keterangan
Pertama	( )	Operator yang terdapat pada tanda kurung akan dihitung lebih dahulu
Kedua	* / %	Operator berikut ini akan dijalankan setelah operator yang terletak di dalam tanda kurung. Apabila terdapat lebih dari satu operator, maka urutan pengerjaannya dilakukan mulai dari kiri sampai kanan
Ketiga	+ -	Operator ini merupakan operator terakhir yang dijalankan oleh <i>PHP</i> setelah operator yang terletak di tanda kurung dan operator * / %. Apabila terdapat lebih dari satu operator, maka urutan pengerjaannya dilakukan mulai dari kiri ke kanan

2. Operator Perbandingan adalah operator yang dilakukan untuk membandingkan antara dua buah *operand*.

Operator	Fungsi	Contoh	Hasil
<	Lebih Kecil	$2 < 7$	True
>	Lebih Besar	$2 > 7$	False
< =	Lebih Kecil atau sama dengan	$2 <= 7$	True
> =	Lebih Besar atau sama dengan	$2 >= 7$	False
= =	Sama dengan	$2 == 7$	False
! = atau <>	Tidak sama dengan	$2 != 7$	True

3. Operator Logika adalah operator yang menyusun kalimat ekspresi / ungkapan logika. Hasil operasi akan didapatkan nilai satu (1) jika bernilai benar dan nol (0) jika salah.

Operator	Fungsi
AND atau &&	Operasi Logika AND (bernilai 1 jika semua input bernilai 1)
OR atau	Operasi Logika OR (bernilai 1 jika salah satu atau semua nilai 1)
XOR	Operasi Logika Eksklusif OR (bernilai 1 jika input nya berbeda)
!	Inkaran atau Negasi (bernilai kebalikan)

Untuk penggunaan operator logika, berikut contoh tabel kebenaran

\$a	\$b	\$a AND \$b	\$a OR \$b	\$a XOR \$b	! (\$a AND \$b)
1	1	1	1	0	0
1	0	0	1	1	1
0	1	0	1	1	1
0	0	0	0	0	1

4. Operator *Bitwise* adalah operator yang membandingkan *binary*. Memungkinkan merubah bit-bit tertentu dalam sebuah bilangan biner. << Shift Left : menggeser bit ke kiri. Contoh:  $7 \ll 2$ . Biner nya  $7 = 111_{bin}$ , digeser kiri sebanyak 2 bit, menjadi  $11100_{bin}$ , sehingga hasilnya  $11100_{bin} = 28$

Operator	Keterangan
>>	Pergeseran bit ke kanan
<<	Pergeseran bit ke kiri
~	Komplemen satu atau NOT
&	Bitwise AND
	Bitwise OR
^	Bitwise XOR (Exclusive OR)

5. Operator *String* adalah operator untuk penggabungan teks. Simbol yang digunakan karakter titik / *point* (.) Berikut *formatstring* beserta keterangannya :

<b>Format</b>	<b>Keterangan</b>
% d	Notasi desimal
% b	Notasi biner
% o	Notasi oktal
% x	Heksa huruf kecil
% X	Heksa huruf besar
% c	Karakter <i>ASCII</i>
% s	<i>String</i>
% f	<i>Double</i> (bilangan real)

## 6. Kombinasi Operator Aritmatik dan *Assignment* (=)

Operator	Contoh	Ekuivalen dengan
+	<code>\$x + = 1;</code>	<code>\$x = \$x + 1;</code>
-	<code>\$x - = 1;</code>	<code>\$x = \$x - 1;</code>
+	<code>\$x + = 2;</code>	<code>\$x = \$x + 2;</code>
-	<code>\$x - = 4;</code>	<code>\$x = \$x - 4;</code>
*	<code>\$x * = 3;</code>	<code>\$x = \$x * 3;</code>
/	<code>\$x / = 2;</code>	<code>\$x = \$x / 2;</code>
%	<code>\$x % = 5;</code>	<code>\$x = \$x % 5;</code>
.	<code>\$my_str .= "hello";</code>	<code>\$my_str = \$my_str . "hello";</code>

7. Operator Ternary merupakan operator menggunakan tanda '?', syntax : (ekspresi1) ? (ekspresi2) : (ekspresi3). Operator ternary menguji ekspresi1, jika hasilnya *TRUE*, maka ekspresi 2 dijalankan, jika hasil ekspresi1 *FALSE*, maka ekspresi3 yang dijalankan. Contoh penggunaan operator dalam *PHP* :

```

<html>
<head><title>Tipe databoolean</title></head>
<body>
<?php
    $a = 17 ;
    $b = 7 ;
    $jumlah = $a+$b ;
    $teks1 = "Aku Sedang Belajar " ;
    $teks2 = "Pemrograman WEB " ;
    $teks3 = "menggunakan PHP";
    printf ("$a > $b = %d <BR> \n", $a > $b); //
    Perbandingan

    echo "Penjumlahan: $a+$b = ".$jumlah."<br>"; //
    Aritmatika
        $a += 1; echo "$a<br/>"; //

```

```

Kombinasi

//===== Operator logika =====
$perbandingan_1 = ($a != $b) ;
$perbandingan_2 = ($a < $b) ;
$logika = $perbandingan_1 OR
$perbandingan_2;
printf(" %d OR %d adalah %d<BR> \n",
$perbandingan_1, $perbandingan_2, $logika);

// ===== OperatorString =====
$gabung_1 = $teks1 . $teks2 . $teks3;
$gabung_2 = $teks1 . " " . $teks2 . " " .
$teks3;
printf(" %s<BR>\n", $gabung_1);
echo $gabung_2 . "<br>";

// ===== Operator ternary =====
$tenary = ($a<5) ? ($a*3) : ($a*4) ;    echo "$tenary
<br>";

// ===== Operator bitwise =====
$bitwise = $a >> 2 ;
echo "\$a digeser sebanyak 2 bit ke kanan
hasilnya adalah $bitwise";

?>
</body>
</html>

```

Hasilnya :

```

Operator
localhost/prakt-wp/pr...
17 > 7 = 1
Penjumlahan: 17+7 = 24
18
1 OR 0 adalah 1
Aku Sedang Belajar Pemrograman WEB menggunakan PHP
Aku Sedang Belajar Pemrograman WEB menggunakan PHP
72
Sa digeser sebanyak 2 bit ke kanan hasilnya adalah 4

```

## D. Soal Latihan (Essay)

1. Pada gambar di bawah, ada berapa banyak tipe data. Sebutkan dan jelaskan tipe data yang digunakan!

```
1 <?php
2 $text1 = "Panjang";
3 $text2 = "Lebar";
4 $text3 = "Luas";
5 $text4 = "Persegi Panjang";
6 $p = 5;
7 $l = 4;
8 $luas = $p * $l;
9 printf("Ada sebuah $text4 dengan
10     <br> $text1 $p cm
11     <br> $text2 $l cm, maka
12     <br> $text3 $text4 adalah " . $luas . " cm");
13
```

2. Pada gambar di bawah terdapat berapa tipe data yang digunakan. Sebut dan jelaskan tipe data yang ada!

0.78  
20  
contoh tipe data

3. Jelaskan perbedaan dari tipe data Float dan *Double*, berikan contohnya masing – masing!

## E. Soal Latihan (Pilihan Ganda)

1. Tipe data yang hanya memiliki 2 nilai yaitu, *true* dan *false* adalah...
  - a. Aritmatika
  - b. *Double*
  - c. *String*
  - d. *Boolean*
2. Berikut ini merupakan tipe data yang dapat digunakan untuk penulisan nama adalah?
  - a. Float
  - b. *Double*
  - c. *Text*
  - d. *Integer*

3. Script php dibawah ini akan menghasilkan :

```
<?php
    $x=10;
    $y=5;
    $z=$x*$y;
    echo"hasil nya adalah <i>$z</i> <b>cm</b><sup>2</sup>";
?>
```

- a. hasil nya adalah *50 cm<sup>2</sup>*
- b. hasil nya adalah 50 cm<sup>2</sup>
- c. hasil nya adalah *50 cm<sup>2</sup>*
- d. hasil nya adalah 50 cm<sup>2</sup>

4. Tipe data yang di gunakan untuk menyimpan bilangan desimal menggunakan...

- a. Float
- b. *Double*
- c. *Text*
- d. *Integer*

5. Script php dibawah ini akan menghasilkan :

```
<?php
    $a=5;

    $b=15;

    $c=$a*$b;

    echo "$b";
?>
```

- a. 45
- b. 10
- c. 15
- d. 20



# BAB 4

## PERCABANGAN

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa percabangan pada *PHP*
2. Mahasiswa dapat mengetahui sintaks penulisan struktur percabangan menggunakan bahasa pemrograman *PHP*.
3. Mahasiswa dapat membuat percabangan dengan *PHP*.

### B. Struktur Percabangan

Percabangan adalah struktur kendali yang berfungsi untuk melakukan pemilihan atas perintah yang akan dijalankan untuk melakukan pemilihan atas perintah yang akan dijalankan sesuai dengan kondisi tertentu. Ada empat perintah percabangan yang akan dibahas di dalam modul ini, yaitu perintah IF, IF-ELSE, ELSE-IF dan SWITCH.

### C. Struktur IF

Pengertian Struktur IF dalam bahasa pemrograman adalah sebuah struktur logika untuk membuat percabangan alur program. Dengan struktur IF dapat diatur apakah sebuah perintah akan dijalankan atau tidak tergantung kepada suatu kondisi.

Struktur logika IF membutuhkan 2 inputan, yaitu ekspresi logika (*expression*) dimana berisi kondisi yang harus dipenuhi, dan perintah yang akan dijalankan (*statement*) jika kondisi logika tersebut terpenuhi. Berikut adalah struktur dasar penulisan alur logika IF dalam *PHP*:

```
if (expression)
{
    Pernyataan yang akan dijalankan apabila kondisi
    benar
}
```

Hasil dari *expression* harus tipe *boolean*. Selama hasil *expression* bernilai *TRUE*, maka *statement* akan dijalankan, namun jika nilainya *FALSE*, maka *statement* tidak akan dijalankan. Aturan Penulisan Struktur IF dalam *PHP* :

1. Penulisan sederhana dari struktur if adalah sebagai berikut :

```
<?php
    if (expression)
        statement;
?>
```

2. Apabila *statement* terdiri dari 2 baris atau lebih

```
<?php
    if (expression)
    {
        statement1;
        statement1;
    }
?>
```

3. Beberapa logika IF sekaligus untuk berbagai situasi

```
<?php
    if (expression1)
    {
        statement1;
        statement2;
    }
    if (expression2)
    {
        statement3;
        statement4;
    }
?>
```

4. Struktur IF didalam IF, atau dikenal dengan nested IF

```
<?php
    if (expression1)
    {
        statement1;
        if (expression2)
        {
            Statement2;
        }
    }
?>
```

Seberapa banyak kondisi IF didalam IF (nested) tidak dibatasi di *PHP*.

#### 5. Alternatif Penulisan Struktur Logika IF

Selain menggunakan tanda kurung kurawal sebagai tanda awal dan akhir IF, *PHP* menyediakan cara penulisan lain untuk menandai akhir perintah IF, yaitu diawali dengan tanda titik dua (:) dengan diakhiri dengan endif.

```
<?php
  if (expression) :
    statement1;
    statement1;
  endif;
?>
```

### D. Struktur Logika IF-ELSE

Jika **Struktur IF** digunakan untuk percabangan alur program dengan 1 pilihan, maka dengan struktur ELSE dapat dibuat percabangan kedua, yakni percabangan ketika kondisi IF tidak terpenuhi, atau expressi IF menghasilkan nilai **FALSE**.

Aturan Penulisan Struktur IF-ELSE dalam *PHP*

#### 1. Penulisan sederhana dari struktur IF-ELSE

```
<?php
  if (expression)
    statement1;
  else
    statement2;
?>
```

#### 2. Struktur logika IF-ELSE terdiri dari beberapa baris

```
<?php
  if (expression)
  {
    statement1;
    statement1;
  }
  else
  {
    statement2;
    statement1;
  }
?>
```

### 3. Penulisan Alternatif Struktur ELSE

```
<?php
  if (expression) :
    statement1;
    statement2;
  else:
    statement3;
  endif;
?>
```

## E. Struktur Logika ELSE-IF

Struktur **ELSE-IF** merupakan percabangan logika lanjutan dari IF. Dengan **ELSE-IF** bisa dibuat kode program yang akan menyeleksi berbagai kemungkinan yang bisa terjadi. Perbedaan IF dengan **ELSEIF**, adalah adanya efisiensi pemrosesan. Untuk kode *PHP* dengan beberapa IF, seluruh kondisi akan dijalankan, walaupun sebenarnya tidak perlu. Lain halnya jika menggunakan perintah **ELSEIF**, maka ketika sebuah kondisi telah dipenuhi, *PHP* tidak perlu melakukan pengecekan terhadap kondisi IF lainnya. Dalam *PHP*, kita bisa menuliskan struktur ELSE-IF dengan `elseif`, atau `else if` (dipisahkan dengan spasi). Kedua bentuk ini dianggap sama.

#### 1. Format dasar penulisan ELSE-IF

Dalam *PHP*, penulisan struktur ELSE-IF bisa ditulis dengan “`elseif`” atau “`else if`” (dipisahkan dengan spasi). Kedua bentuk ini dianggap sama.

```
<?php
  if (expression) {
    statement1;
  }
  else if {
    statement2;
  }
  else {
    statement3;
  }
?>
```

#### 2. Penulisan Alternatif Struktur ELSE-IF

Namun untuk cara penulisan ini, tidak bisa memisahkan penulisan ELSE-IF menjadi “`else if`”, tetapi harus ditulis menyatu menjadi “`elseif`”.

```
<?php
    if (expression) :
        statement1;
        statement2;
    elseif (expression):
        statement3;
    else :
        statement4;
    endif;
?>
```

## F. Struktur Logika *Switch*

**Struktur** logika *switch* adalah sebuah stuktur percabangan yang akan memeriksa suatu variabel, lalu menjalankan perintah-perintah yang sesuai dengan kondisi yang mungkin terjadi untuk variabel tersebut. Struktur *switch* ini mirip dengan struktur IF yang ditulis berulang. Aturan Penulisan Struktur *Switch* dalam *PHP* :

```
<?php
    switch ($var)
    {
        casevalue1:
            statement1;
            break;
        casevalue2:
            statement2;
            break;
    }
?>
```

### **Perbedaan Antara struktur IF dengan Switch :**

Didalam struktur *switch*, kondisi logika hanya akan diperiksa satu kali saja, yaitu pada awal perintah *switch*, dan hasilnya di bandingkan dengan setiap *case*. Akan tetapi di dalam struktur *if*, setiap kondisi akan selalu diperiksa. Sehingga jika anda memiliki struktur percabangan yang banyak, struktur *switch* akan lebih cepat dieksekusi.

Namun disisi lain, *switch* memiliki keterbatasan dalam jenis operasi perbandingan yang dapat dilakukan. Operasi perbandingan di dalam *switch* terbatas untuk hal-hal sederhana seperti memeriksa nilai dari sebuah variabel. Struktur *switch* tidak bisa digunakan untuk percabangan program dengan operasi yang lebih rumit seperti membandingkan 2 variabel. *Switch* tidak bisa digunakan untuk membuat kode program untuk menentukan nilai terbesar.

## G. Soal Latihan (Essay)

1. Buatlah Script percabangan berikut :

Joko mendapat nilai 79, jika nilai  $0 > 20$  mendapatkan grade E, nilai  $21 > 40$  mendapatkan grade D, nilai  $41 > 60$  mendapatkan grade C, nilai  $61 > 80$  mendapatkan grade B, dan nilai  $81 > 100$  mendapatkan grade A. Maka Hasil nilai joko mendapatkan grade?

2. Lengkapi Program di bawah agar dapat berjalan tanpa *error*.

```
<?php
echo "X=10 <br>";
echo "jika X lebih besar dari 12 maka salah dan
      jika x kurang dari sama dengan 10 maka benar"; echo "<br>";
$x = 10;
if($x .. ..) {
    echo ".....";
}
else if($x .. ..) {
    echo ".....";
}
?>
```

3. Buatlah script yang sesuai untuk melengkapi script *PHP* dibawah ini agar dapat menghasilkan output seperti tampilan di samping!

```
<?php
$uang = 11500;
$jumlah = 2;
.....
switch(.....){
    .....
}
echo "Nama barang : $beli <br>";
echo "Harga : $harga <br>";
echo "Jumlah : $jumlah <br>";
echo "Total harga : $total <br>";
echo "$hasil";
?>
```

Nama barang : Penggaris
Harga : 3500
Jumlah : 2
Total harga : 7000
Kembali : 4500

## H. Soal Latihan (Pilihan Ganda)

1. Script *PHP* di bawah ini menghasilkan :

```
<?php
    $nilai=67;
    if($nilai<21){
        echo"nilai $nilai menjadi A";
    }
    else if($nilai<41){
        echo"nilai $nilai menjadi B";
    }
    else if($nilai<61){
        echo"nilai $nilai menjadi C";
    }
    else if($nilai<81){
        echo"nilai $nilai menjadi D";
    }
    else{
        echo"nilai $nilai menjadi E";
    }
?>
```

- Nilai 67 menjadi D
- Nilai 67 menjadi C
- Nilai 67 menjadi B
- Nilai 67 menjadi A

2. Soal :

```
<?php
    $uang = .....;
    $sepatu = 225000;
    $tas = 185000;
    $kemeja = 150000;
    $celana = 215500;
    $total = $sepatu + $tas + $kemeja + $celana;

    if($uang >= $total):
        echo "Anda dapat membeli semua barang";
    elseif($uang >= $total-$sepatu):
        echo "Anda dapat membeli Kemeja, Tas, dan Celana";
    elseif($uang >= $total-( $celana+$sepatu)):
        echo "Anda dapat membeli Kemeja dan Tas";
    elseif($uang >= $kemeja && $uang < $tas+$kemeja):
        echo "Anda hanya dapat membeli satu barang";
    endif;
?>
```

Jika variabel uang diisi dengan nilai 420500, manakah output yang akan ditampilkan?

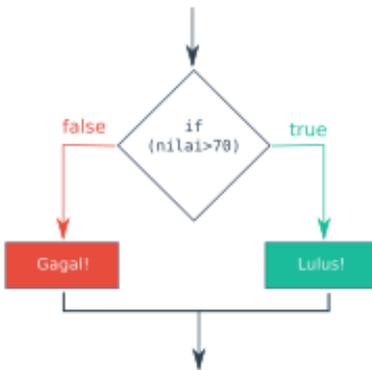
- Anda dapat membeli semua barang
- Anda dapat membeli Kemeja, Tas, dan Celana
- Anda dapat membeli Kemeja dan Tas
- Anda hanya dapat membeli satu barang

3. Lengkapi Script (...) di bawah :

```
1 <?php
2
3 $umur = 13;
4
5 if (... < 18 ){
6     echo "Kamu tidak boleh membuka situs ini!";
7 } else {
8     echo "Selamat datang di website kami!";
9 }
10
```

- a. ?>
- b. Umur
- c. \$umur
- d. 13

4. Dari Flow Chart di bawah ini jika user menginputkan nilai 60 maka hasil dari percabangan adalah :



- a. Gagal!
- b. Lulus!
- c. Mengulang!
- d. Semua salah

5. Lengkapi Script di bawah ini agar memunculkan **echo LULUS**

```
<?php
$nilai = 70;
$nilai2 = 89;
if (... > 80){
    echo "lulus";
}
else{
    echo "gagal";
}
?>
```

- a. \$nilai
- b. 70
- c. \$nilai2
- d. 89

# BAB 5

## PERULANGAN

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa perulangan pada *PHP*
2. Mahasiswa dapat mengetahui sintaks penulisan struktur perulangan menggunakan bahasa pemrograman *PHP*.
3. Mahasiswa dapat membuat perulangan dengan *PHP*.

### B. Struktur Perulangan

Struktur perulangan (atau dalam bahasa Inggris disebut dengan *loop*) adalah instruksi program yang bertujuan untuk mengulang beberapa baris perintah. Dalam merancang perulangan kode program, setidaknya harus ada 3 komponen, yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir dimana perulangan akan berhenti. Struktur perulangan di *PHP* terdiri dari 3 jenis perintah perulangan yaitu *For*, *While*, dan *Do-While*.

### C. Struktur Perulangan For

Struktur *For* digunakan untuk mengulangi suatu perintah dengan jumlah pengulangan yang sudah diketahui. Pada perintah ini tidak perlu menuliskan suatu kondisi yang diuji. Cara penulisan *script* untuk struktur *For* :

Cara 1 :

```
<?php
    for (start; cond; counter)
    { statement;
    }
?>
```

## Cara 2 :

```
<?php
    for (start; cond; counter):
        statement;
        endfor ;
?>
```

1. *Start* adalah inisialisasi atau nilai pada saat awal perulangan.
2. *Cond* merupakan kondisi yang akan membatasi perulangan.
3. *Counter* merupakan pertambahan atau pengurangan nilai dari variabel sehingga perulangan tetap berjalan.
4. *Statement* adalah bagian kode program yang akan diproses secara terus-menerus selama proses perulangan berlangsung.

*Infinity Loop* : perulangan yang berjalan terus menerus, karena kondisi akhir tidak pernah terpenuhi. *Nested Loop* (Perulangan Bersarang) : membuat perulangan di dalam perulangan.

```
<html>
<head>
<title>Perulangan dengan FOR</title>
</head>
<body>
<?php
    For($i=1; $i<=7; $i++)
    {
        Print("<h$i>Ini contoh penggunaan FOR$h$i</h$i>");
    }
</body>
?>
```

## D. Struktur Perulangan *While*

Struktur perulangan *while* digunakan untuk kondisi perulangan dimana banyaknya perulangan tidak dapat dipastikan pada saat penulisan program. Perulangan akan terus berjalan selama kondisi masih bernilai benar.

### Cara 1 Penulisan Struktur Perulangan *While* :

```
<?php
    init_awal;
    while (condition)
    {
        statement;
        counter;
    }
?>
```

## Cara 2 Penulisan Struktur Perulangan *While* :

```
<?php
    init_awal;
    while (condition):
        statement;
        counter;
    endwhile;
?>
```

1. *init\_awal* merupakan inisialisasi atau nilai awal *variable*.
2. *Condition* adalah kondisi yang akan membatasi perulangan.
3. *Statement* adalah kode program yang akan diulang.
4. *counter* merupakan penambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.
5. Tanda kurung kurawal : untuk membatasi blok program yang akan diulang.

## E. Struktur Perulangan *Do - While*

Pada perulangan *do-while*, pengecekan kondisi dilakukan di akhir perulangan, sehingga walaupun kondisi *FALSE*, perulangan akan tetap berjalan minimal 1 kali.

Cara Penulisan Struktur Perulangan *Do-While* :

```
<?php
    init_awal;
    do {
        statement;
        counter;
    } while (condition);
?>
```

Perbedaan antara perintah *While* dengan *Do-While* adalah letak dari kondisi yang diperiksa. Pada perintah *While*, kondisi yang diperiksa terletak di awal perulangan, sehingga sebelum masuk ke dalam perulangan *While* kondisi harus benar. Sedangkan pada *Do-While*, kondisi diperiksa di akhir perulangan. Ini berarti paling sedikit sebuah perulangan akan dilakukan oleh perintah *Do-While*, karena untuk masuk ke perulangan tidak ada kondisi yang harus dipenuhi.

## F. *Break* Dalam Perulangan

Perintah *break* akan menyebabkan program menghentikan perulangan (langsung keluar dari perulangan). Perintah *break* dapat di letakkan di posisi manapun di dalam perulangan, namun biasanya logika IF digunakan untuk menentukan kapan perintah *break* dijalankan.

## G. *Continue* Dalam Perulangan

Perintah *continue* akan melanjutkan proses perulangan tanpa melewati (meng-eksekusi) baris perintah di bawahnya. Perintah *continue* akan menghentikan perulangan untuk 1 iterasi saja, lalu proses perulangan akan dilanjutkan.

## H. Soal Latihan (Essay)

1. Buatlah Script berikut menggunakan pengulangan **for**

```
Akan menampilkan perulangan for ke 6  
Akan menampilkan perulangan for ke 7  
Akan menampilkan perulangan for ke 8  
Akan menampilkan perulangan for ke 9  
Akan menampilkan perulangan for ke 10
```

2. Jelaskan penggunaan fungsi *Break* dan *Continue* dalam kasus perulangan, dan tuliskan contoh penggunaannya!
3. Buatlah script *PHP* untuk membuat tampilan di bawah ini :

```
1  
22  
333  
4444  
55555  
666666  
7777777
```

## I. Soal Latihan (Pilihan Ganda)

1. Berikut ini merupakan jenis perulangan, manakah yang bukan jenis perulangan dari...
  - a. For
  - b. *While* - do
  - c. Do - while
  - d. For - while
2. Perulangan akan berjalan terus menerus, karena kondisi akhir tidak pernah terpenuhi. Kalimat tersebut merupakan definisi dari?
  - a. Nested Loop
  - b. Infinity Loop
  - c. *Break* dalam Loop
  - d. Continue dalam Loop
3. Berdasarkan gambar di bawah ini. Jika variabel nilai1 = 7 dan nilai2 = 75, maka outputnya adalah:

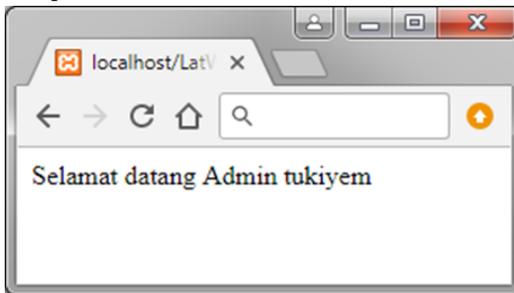
```
<?php
function bil_prima($a, $b){
    for($a;$a<=$b;$a++){
        $bil = true;
        for($i=2;$i<$a;$i++){
            if($a%$i == 0){
                $bil = false;
            }
        }
        if($bil){
            echo "$a ";
        }
    }
}
$nilai1 = ?;
$nilai2 = ?;
bil_prima ($nilai1, $nilai2);
?>
```

- a. 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73

- b. 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43  
45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75
- c. 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44  
46 48 50 52 54 56 58 60 62 64 66 68 70 72 74
- d. 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61  
64 67 70 73

4. Looping (Perulangan) dalam php, angka *default* di mulai dari angka...
- a. 5
  - b. 3
  - c. 1
  - d. 0

5. Output :



Manakah script yang sesuai untuk melengkapi script *PHP* diatas agar muncul output seperti tampilan diatas?

a.	<pre> if(\$user['id']==\$status['id']){     echo "Selamat datang ".\$status['status'].     ".\$user['username']; } </pre>
b.	<pre> for(\$a=0; \$a&lt;count(\$status); \$a++){ if(\$user['id']==\$status[\$a]['id']){     echo "Selamat datang     ".\$status[\$a]['status'].     ".\$user['username']; } } </pre>

c.	<pre> for(\$a=0; \$a&lt;count(\$status); \$a++){   if(\$user['id']==\$status[\$a]['stats']['id'){     echo "Selamat datang     ".\$status[\$a]['stats']['status'].     ".\$user['username'];   } } </pre>
d.	<pre> if(\$user['id']==\$status[1]['stats']['id'){   echo "Selamat datang   ".\$status[1]['stats']['status'].   ".\$user['username']; } </pre>



# BAB 6

## ARRAY

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa *array* pada *PHP*
2. Mahasiswa dapat mengetahui sintaks penulisan struktur *array* menggunakan bahasa pemrograman *PHP*.
3. Mahasiswa dapat membuat *array* dengan *PHP*.

### B. Array

*Array* merupakan sekumpulan struktur data yang dapat menyimpan banyak nilai ataupun elemen dalam sebuah variabel secara berurutan. Data pada *array* tersimpan dengan menggunakan indeks untuk memudahkan pencarian kembali data tersebut. Berbeda dengan variabel yang hanya dapat menyimpan satu nilai, *array* mampu menampung sejumlah nilai. *Array* akan memudahkan dalam membuat kelompok data, serta menghemat penulisan dan penggunaan variabel.

### C. Pendeklarasian Array

*Array* sebagai ruang penyimpanan data yang memiliki nilai-nilai yang bisa dipanggil dengan angka atau indexnya. Komponen *array* terdiri dari pasangan index (kunci / *key*) dan elemen (nilai / *value*). *Key* adalah penunjuk posisi dimana *value* disimpan. Semua elemen *array* yang tersimpan memiliki tipe data yang sama. *Array* dapat berdimensi satu, dua, tiga atau lebih. *Array* berdimensi satu (*one-dimensionalarray*) mewakili bentuk suatu vektor. *Array* berdimensi dua (*two-dimensionalarray*) mewakili suatu matrix. *Array* berdimensi tiga (*three-dimensionalarray*)

## D. IndexedArray tanpa Key

*Indexedarray* berarti bahwa *array* memiliki *key* berupa angka yang berurutan. Secara *default* *value1* memiliki *key* 0, *value2* memiliki *key*1, dst. Penulisan *Indexedarray* :

```
$nama_variabel [] = "value1";  
$nama_variabel = array("value1", "value2", "dst");  
$nama_variabel = ["value1", "value2", "dst"]; // mulai PHP  
5.4
```

## E. IndexedArray dengan Key

*Indexedarray* dengan *key* didefinisikan, sehingga *value1* tidak pasti *key* nya 0. Penulisan *Indexedarray* dengan *key* :

```
$nama_variabel [1] = "value1";  
$nama_variabel = array(1=>"value1", 2=>"value2",  
3=>"value3");  
$nama_variabel = [2=>"value1", 3=>"value2", 4=>"value3" ];  
$nama_variabel = ['value1', 3=>'value2', 'value3',  
'value4'];
```

## F. Associative Array

Selain menggunakan angka, *key* dalam *PHP* dapat berisi *string* atau *boolean*. Sedangkan untuk *value* dapat menyimpan berbagai tipe data seperti *integer*, *float*, *string*, *boolean*, bahkan *array* lainnya. *Array* seperti ini disebut juga dengan istilah *associativearray*, berarti bahwa kita mendefinisikan sendiri *key* dari *array* tersebut. Penulisan *associativearray* :

```
$nama_variabel = array(key1=>value1, key2=>value2 );  
$nama_variabel = [key1=>value1, key2=>value2];
```

## G. Array Multidimensi

*Array* bisa memiliki kedalaman yang tidak terbatas, yang sering disebut *array* multidimensi, bisa dua dimensi, tiga, dst... Multidimensi disini dapat diibaratkan *array* didalam *array*, maksudnya *value* dari

suatu *array* menjadi index dari *array* yang lain, sehingga *value* ini bentuknya juga *array*.

Contoh Penulisan *Array* Multidimensi :

```
$variabel = [  
    [key1 => [key1a =>value1a, key1b =>value1b]],  
    [key2 => [value2a, value2b, value2c]]  
];
```

## H. Menampilkan *ValueArray*

Untuk mengakses *value* dari *array*, digunakan *key* dari *value* yang penulisannya diapit dengan kurung siku [ ]. Perintah `print_r` digunakan untuk menampilkan seluruh isi *array* beserta *key* nya

```
echo $variabel [key1];  
echo $variabel [key2];  
print_r($variabel);
```

Cara lainnya adalah menggunakan perulangan *foreach* :

```
foreach ($variabel as $value)  
{  
    statement (...$value...)  
}
```

Atau

```
foreach ($variabel as $key => $value)  
{  
    statement ($key...$value...)  
}
```

## I. Menambahkan Elemen *Array*

Untuk menambah element pada *array*, bisa kita lakukan dengan menggunakan tanda kurung siku, baik berisi *key* maupun tidak. Penambahan element pada *IndexedArray* :

```
$variabel [1] = 'value1'; // Dengan mendefinisikan  
index: 1  
$variabel [] = 'value2'; // PHP akan melanjutkan ke index  
berikutnya,  
$variabel [] = 'value3'; // yaitu 2, 3, dst.
```

## Penambahan element pada Associative Array :

```
$variabel = ['key1' => 'value1', 'key2' => 'value2',  
'key3' => 'value3'];  
$variabel ['key4'] = 'value4';
```

## J. Menghapus Elemen Array

Terdapat dua fungsi untuk menghapus element dari *array*, yaitu fungsi `unset` dan fungsi `array_slice`. Fungsi `unset` digunakan jika *key* dari *array* sudah diketahui. Fungsi `array_slice` digunakan dengan cara mengambil beberapa bagian dari *array* dan menghapus sisanya. Contoh penulisan fungsi `unset` :

```
$variabel = ['key1' => 'value1', 'key2' => 'value2',  
'key3' => 'value3'];  
unset($variabel ['key2']);
```

## Contoh penulisan fungsi `array_slice` :

```
$variabel_a = ['value1', 'value2', 'value3', 'value4'];  
$variabel_b = array_slice($variabel_a, 1, 2);  
// 1 : mulai awal key yang akan diambil  
// 2 : jumlah key yang akan diambil
```

## K. Soal Latihan (Essay)

1. Buatlah sebuah gambaran mengenai *array* 1 dimensi, 2 dimensi, dan 3 dimensi!
2. Buatlah Script *PHP* untuk tampilan seperti di bawah ini menggunakan *array* di *PHP*!

```
asus [1] => lenovo [2] => Iphone [3] => Oppo [4] => Nokia )
```

3. Tuliskan contoh *array* dengan *key* / *array\_key()* !

## L. Soal Latihan (Pilihan Ganda)

1. Apakah Output dari script di bawah ini?

```
<?php
$array = array ('3' => 'a', '1b' => 'b', 'c', 'd');
echo ($array[1]);
?>
```

- a. 1
  - b. b
  - c. c
  - d. warning
2. Apakah Output dari script di bawah ini?

```
<?php
$array = array (0.5 => 'a', 1 => 'b');
echo count ($array);
?>
```

- a. 1
  - b. 2
  - c. 0
  - d. 1.5
3. Yang dimaksud dengan *array* multidimensi adalah...
    - a. **Array** yang bisa di akses dengan satu buah indeks
    - b. **Array** di dalam *array*
    - c. Tempat untuk menyimpan data pada program yang kalian jalankan
    - d. Kumpulan data atau nilai yang berkumpul dalam satu *array*

4. Berikut ini merupakan jenis *array* dalam *PHP* adalah ?
  - a. *Associative Array*
  - b. *Normalized Array*
  - c. *Constructive array*
  - d. *General Array*

5. Script *PHP* :

```
<?php
$epl[3]="EPL adalah English Premier League";
$epl[90]="EPL menjadi liga yang paling banyak ditonton di dunia";
$epl[77]="Persaingan yang kompetitif menjadi daya tarik utama liga ini";
$epl[38]="Pemain-pemain top dunia juga mempunyai ambisi bermain di EPL";
$epl[54]="dan memenangkan piala EPL";

echo"<pre>";
print_r($epl);
echo"</pre>";
?>
```

Gambar yang tepat dari script *PHP array* di bawah ini adalah...

- a. 

```
Array
(
    [3] => EPL adalah English Premier League
    [90] => EPL menjadi liga yang paling banyak ditonton di dunia
    [38] => Pemain-pemain top dunia juga mempunyai ambisi bermain di EPL
    [77] => Persaingan yang kompetitif menjadi daya tarik utama liga ini
    [54] => dan memenangkan piala EPL
)
```
- b. 

```
Array
(
    [90] => EPL menjadi liga yang paling banyak ditonton di dunia
    [77] => Persaingan yang kompetitif menjadi daya tarik utama liga ini
    [54] => dan memenangkan piala EPL
    [38] => Pemain-pemain top dunia juga mempunyai ambisi bermain di EPL
    [3] => EPL adalah English Premier League
)
```
- c. 

```
Array
(
    [3] => EPL adalah English Premier League
    [38] => Pemain-pemain top dunia juga mempunyai ambisi bermain di EPL
    [54] => dan memenangkan piala EPL
    [77] => Persaingan yang kompetitif menjadi daya tarik utama liga ini
    [90] => EPL menjadi liga yang paling banyak ditonton di dunia
)
```
- d. 

```
Array
(
    [3] => EPL adalah English Premier League
    [90] => EPL menjadi liga yang paling banyak ditonton di dunia
    [77] => Persaingan yang kompetitif menjadi daya tarik utama liga ini
    [38] => Pemain-pemain top dunia juga mempunyai ambisi bermain di EPL
    [54] => dan memenangkan piala EPL
)
```

# BAB 7

## FUNGSI

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa fungsi pada *PHP*
2. Mahasiswa dapat mengetahui sintaks penulisan struktur fungsi menggunakan bahasa pemrograman *PHP*.
3. Mahasiswa dapat membuat fungsi dengan *PHP*.

### B. Fungsi

Dalam pembuatan program terdapat sekumpulan perintah yang akan dieksekusi berulang kali, hal tersebut akan memakan waktu lama karena harus melakukan ketik ulang perintah ketika akan digunakan. Hal ini dapat diatasi dengan adanya subrutin yang merupakan sekumpulan perintah yang diberi nama dan kemudian dapat kita panggil sewaktu-waktu ketika akan digunakan. Dalam pemrograman yang termasuk kedalam subrutin adalah prosedur dan fungsi.

Perbedaan antara keduanya yaitu : fungsi mengembalikan nilai tertentu, sedangkan prosedur tidak. Bahasa *PHP* hanya mengenal perintah untuk membuat fungsi, yaitu *Function*. Di dalam subrutin terdapat argumen untuk menyimpan nilai-nilai yang akan dipanggil. Prosedur dapat dibuat tanpa menggunakan perintah *function*, yaitu dengan perintah *return*.

### C. Fungsi Dengan Argumen

Argument adalah suatu nilai yang dimasukkan ke dalam sebuah fungsi. Secara *default* sebuah argumen bersifat *pass by values*, yang berarti hanya nilainya saja yang dibutuhkan sehingga nilai pada variabel tersebut tidak mengalami perubahan setelah fungsi dijelaskan.

Selain argumen yang telah dijelaskan, ada argumen yang bersifat *pass by reference*, yang digunakan ketika kita ingin mengubah nilai sebuah argumen. Argumen yang bersifat *pass by reference* ini nilainya akan berubah setelah fungsi dijalankan. Untuk membuat sebuah argumen *pass by reference* kita akan menggunakan karakter “&” di depan nama argumen.

## D. Fungsi Built-in

Fungsi built-in membahas tentang seluruh fungsi yang telah disediakan oleh *PHP*.

### 1. Fungsi Umum dalam *PHP* :

<code>echo ( )</code>	: mengirim satu atau lebih parameter yang dipisahkan dengan tanda baca koma (,) ke <i>browser</i>
<code>print ( )</code>	: mengirim output ke <i>browser</i>
<code>printf ( )</code>	: mengirim output ke <i>browser</i> , dan dengan fungsi tersebut dapat untuk menampilkan hasil dengan <i>format</i> yang dapat diatur
<code>copy ( )</code>	: Menyalin suatu <i>file</i> yang ditentukan dalam argumen <i>source</i> (lokasi asal <i>file</i> ) menuju ke <i>dstination</i> (lokasi tujuan). Akan tetapi menghasilkan <i>true</i> jika berhasil
<code>include ( )</code>	: Digunakan untuk menyisipkan <i>file</i> yang disebut dalam argumen
<code>phpinfo ( )</code>	: Mengambil informasi tentang versi <i>PHP</i> , pembuatnya, sistem operasi pada <i>webserver</i> , konfigurasi variabel dan sebagainya.
<code>strlen ( )</code>	: Mengembalikan nilai <i>integer</i> yang merupakan panjang <i>string</i> dalam argumen <i>text</i>
<code>ord ( )</code>	: Mengembalikan nilai <i>ASCII</i> dari karakter pada argumen. Argumen yang digunakan adalah karakter.
<code>Strtolower ( )</code>	: Mengubah argumen menjadi huruf kecil semua
<code>strtoupper ( )</code>	: Mengubah argumen <i>test</i> yang bertipe <i>string</i> menjadi huruf kapital

### 2. Fungsi Build-In terkait dengan *Array*

<code>sort &amp; rsort ( )</code>	: Pengurutan <i>array</i> berdasarkan <i>value</i> secara ascending dan descending dengan mengubah <i>index/key</i>
<code>asort &amp; arsort ( )</code>	: Mengurutkan <i>value</i> dari <i>array</i>
<code>krsort &amp; krsort ( )</code>	: Mengurutkan <i>key</i> suatu <i>array</i>

)	
shuffle ( )	: Mengurutkan <i>key</i> secara random
current ( )	: Mendapatkan elemen <i>array</i> yang ditunjuk oleh <i>pointer</i>
end ( )	: <i>Pointer</i> menunjuk pada elemen <i>array</i> terakhir
key ( )	: Mendapatkan <i>key</i> yang ditunjuk oleh <i>pointer</i>
next ( )	: <i>Pointer</i> menunjuk pada elemen selanjutnya
prev ( )	: <i>Pointer</i> menunjuk pada elemen sebelumnya
reset ( )	: Memindahkan <i>pointer array</i> ke awal (elemen pertama)
array_keys ( )	: Mengambil semua <i>key</i> pada <i>array</i>
array_values ( )	: Mengambil semua <i>value</i> dari <i>array</i>
count ( )	: Menghitung jumlah <i>array</i>
join ( )	: Menggabungkan <i>value</i> pada <i>array</i> menjadi <i>string</i>
array_map ( )	: Mengubah semua <i>value</i> pada <i>array</i>
in_array ( )	: Cek apakah <i>array</i> mengandung <i>value</i> tertentu
key_exists ( )	: Cek apakah <i>array</i> mengandung <i>key</i> tertentu
array_merge	: Menggabungkan <i>array</i> pada <i>PHP</i>
array_search	: Mencari index dari <i>value</i> tertentu
array_shift	: Menghapus element pertama pada <i>array</i>
array_unshift	: Menambah element pertama pada <i>array</i>
array_pop	: Menghapus element terakhir pada <i>array</i>
array_push	: Menambah element terakhir pada <i>array</i>
array_reverse	: Membalik Urutan <i>Array</i>
min & max	: Mencari Nilai Minimal dan Maksimal dari Suatu <i>Array</i>

## E. Fungsi String

*PHP* menyediakan lebih dari 90 fungsi untuk manipulasi *string*. Fungsi-fungsi *string* dalam *PHP* bisa dilihat di <http://ca.php.net/manual/en/ref.strings.php> Beberapa fungsi manipulasi *string* yang sering digunakan, antara lain :

echo ( )	: menampilkan satu atau beberapa <i>string</i> .
strlen ( )	: menghitung panjang <i>string</i>
strtolower ( )	: mengubah <i>string</i> menjadi huruf kecil ( <i>lower-case</i> ).
strtoupper ( )	: mengubah <i>string</i> menjadi huruf kapital ( <i>upper-case</i> )
ucfirst ( )	: mengubah huruf pertama dari semua <i>string</i> menjadi <i>upper-case</i> .

<code>ucwords()</code>	: mengubah huruf pertama tiap kata dalam <i>string</i> menjadi upper-case.
<code>strip_tags()</code>	: mengabaikan <i>HTML</i> dan <i>PHPtag</i> dalam <i>string</i>
<code>htmlentities()</code>	: mengubah semua <i>taghtml</i> menjadi <i>HTML entities</i>
<code>htmlspecialchars()</code>	: mengubah spesial karakter menjadi <i>HTML entities</i>
<code>chr()</code>	: menghasilkan karakter dari bilangan <i>ASCII</i>
<code>ord()</code>	: menghasilkan kode <i>ASCII</i> dari suatu karakter
<code>crypt()</code>	: menghasilkan <i>string</i> terenkripsi searah
<code>md5()</code>	: menghasilkan <i>string</i> terenkripsi searah
<code>number_format()</code>	: mengatur <i>format</i> angka
<code>wordwrap()</code>	: memotong karakter dalam <i>string</i> dengan <i>stringbreak-character</i>
<code>implode()</code>	: menggabungkan elemen <i>array</i> menjadi <i>string</i> dg separator.
<code>explode()</code>	: memecah <i>string</i> berdasarkan suatu delimiter (separator)
<code>split()</code>	: sama dengan <code>explode()</code>
<code>join()</code>	: sama dengan <code>implode()</code>
<code>strpos()</code>	: mencari posisi pertama sebuah <i>string</i> dalam <i>string</i>
<code>stripos()</code>	: mencari posisi pertama sebuah <i>string</i> bersifat <i>case-insensitive</i> .
<code>strrpos()</code>	: mencari posisi terakhir sebuah <i>string</i> dalam <i>string</i>
<code>strstr()</code>	: mengambil <i>string</i> dari posisi karakter pertama dalam <i>string</i>
<code>strrchr()</code>	: mengambil <i>string</i> dari posisi karakter terakhir dalam <i>string</i>
<code>strrev()</code>	: membalik <i>string</i>
<code>substr()</code>	: memotong <i>string</i>
<code>stripslashes()</code>	: menghilangkan backslashes (\) dalam <i>string</i>
<code>addslashes()</code>	: menambahkan backslashes (\) di setiap tanda kutip (quote)
<code>trim()</code>	: menghapus karakter (" ", \t, \n, \r, \0, \x0B) di awal dan akhir <i>string</i>
<code>ltrim()</code>	: menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di awal <i>string</i>
<code>rtrim()</code>	: menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir <i>string</i>
<code>nl2br()</code>	: menambahkan <i>HTML line break</i> ( ) pd baris baru dalam <i>string</i>
<code>str_repeat()</code>	: mengulang <i>string</i>
<code>str_replace()</code>	: mengganti semua <i>string</i> dalam pola menjadi suatu <i>string</i>

## F. Fungsi Date

Tanggal yang akan ditampilkan dalam halaman *web* menyesuaikan waktu *server*, bukan waktu yang ada di client. Hal ini disebabkan *PHP* merupakan *server side programming*. Fungsi *date* digunakan untuk menampilkan data tanggal sesuai dengan *format* yang diinginkan. *Format* penulisannya :

```
date(format_tanggal, timestamp);
```

Argumen kedua, yaitu *timestamp*, sifatnya opsional, jika tidak didefinisikan maka akan digunakan *timestamp* waktu sekarang (tanggal dan waktu pada komputer *server* ketika script dieksekusi). Lebih jauh tentang *timestamp* di bahas dibagian bawah.

*format\_tanggal* berbentuk *string* (teks) yang diwakili oleh karakter tertentu yang mewakili tanggal dan waktu, sehingga, jika *string* tersebut mengandung karakter tersebut, maka akan langsung diterjemahkan menjadi waktu, contoh:

```
<?php echo date('d-m-Y H:i:s'); // Hasil: 26-12-2017  
09:30:15
```

Pada contoh diatas, karena spasi dan dash (-) **tidak** termasuk karakter *date*, maka dibiarkan apa adanya. Jika kita ingin karakter *date* tidak diterjemahkan, maka kita escape karakter tersebut dengan *backslash*. Misal:

Pada contoh diatas, jika tidak diescape, maka hasil yang diperoleh adalah: CET1Tuesday: 26-12-2017 09:30:15

```
<?php  
echodate('\T\g\l: d-m-Y H:i:s');// Tgl: 26-12-2019  
09:30:15
```

*Format* tanggal dan waktu dibuat mengikuti standar Amerika, seperti penggunaan nama hari, nama bulan, penggunaan AM dan PM, dll.

No.	Kode	Keterangan	Contoh nilai
<i>Hari</i>			
1.	l	Nama hari lengkap dalam seminggu	Sunday s/d Saturday
2.	D	Tiga digit nama hari dalam seminggu	Sun s/d Sat

3.	w	Urutan hari dalam seminggu	0 (Sunday) s/d 6 (Saturday)
4.	N	Urutan hari dalam seminggu	1 (Monday) s/d 7 (Sunday)
5.	z	Urutan hari dalam setahun	0 s/d 365
<i>Tanggal</i>			
6.	d	Tanggal, 2 digit dengan nol	01 s/d 31
7.	j	Tanggal tanpa nol	1 s/d 31
8.	S	Akhiran angka English untuk tanggal,	st, nd, rd, th. (2 karakter)
<i>Minggu</i>			
9.	W	Urutan minggu dalam setahun	42 (minggu ke-42)
<i>Bulan</i>			
10.	F	Nama bulan lengkap	January s/d December
11.	M	Tiga digit nama bulan dalam	Jan s/d Dec
12.	m	Urutan bulan dalam setahun dg nol	01 s/d 12
13.	n	Urutan bulan dalam setahun tanpa 0	1 s/d 12
14.	t	Jumlah hari dalam tiap bulan	28, 29, 30, 31
<i>Tahun</i>			
15.	Y	<i>Format</i> tahun 4 digit	2007, 2017
16.	y	<i>Format</i> tahun 2 digit	07, 17
17.	L	1 jika melompati tahun, 0 jika tidak	
<i>Waktu</i>			
18.	A	Uppercase Ante atau Post Meridiem	AM atau PM
19.	a	Lowercase Ante atau Post Meridiem	am atau pm
20.	G	Jam <i>format</i> 24 tanpa nol	0 s/d 23
21.	g	Jam <i>format</i> 12 tanpa nol	1 s/d 12
22.	H	Jam <i>format</i> 24 dengan nol	00 s/d 23
23.	h	Jam <i>format</i> 12 dengan nol	01 s/d 12
24.	i	Menit dengan nol	00 s/d 59
25.	s	Detik dengan nol	00 s/d 59

## G. Penggunaan Timestamp

Timestamp merupakan suatu istilah yang merujuk pada waktu dengan *format* unix timestamp. Pada *PHP*, setidaknya ada dua fungsi yang dapat digunakan untuk membuat timestamp, yaitu : `mktime()` dan `strtotime()`. Function `mktime ( )` digunakan untuk membuat timestamp, dengan sintaks:

```
mktime (hour, minute, second, month, day, year)
```

Function `strtotime ( )` digunakan untuk membuat timestamp, dengan sintaks:

```
strtotime(string, timestamp);
```

*Format* ini mendefinisikan waktu berdasarkan detik yang mengacu pada waktu tertentu, yaitu 01-01-1970 00:00:00, dengan zona waktu GMT+0. Dengan demikian, nilai 1 pada timestamp berarti satu detik sejak 01-01-1970 00:00:00 atau 01-01-1970 00:00:01, perhatikan contoh berikut:

```
<?php date_default_timezone_set('UTC');
echostrtotime("1970-01-01 00:00:01");// Hasil 1
```

Jika dibalik, yaitu dengan mengisikan angka 1 pada argumen ke dua dari fungsi `date()`, maka akan memperoleh hasil yang sama, yaitu 1970-01-01. Kunci dari fungsi ini adalah pada argumen timestamp, jika kosong, maka akan ditampilkan tanggal dan waktu sekarang.

## H. Fungsi Time

Fungsi `time()` digunakan untuk mengambil waktu sekarang pada komputerserver dalam *format* timestamp. Fungsi ini tidak memiliki argumen. Timestamp yang dihasilkan oleh fungsi `time()` selalu pada waktu UTC (GMT + 0), meskipun konfigurasi zone waktu berubah ubah.

```
<?php
echo 'Date: ' . date('Y-m-d H:i:s'); // Hasil: 2017-01-17
15:36:12 (GMT + 1)
echo 'Timestamp: ' . time(); // Hasil: 1484663772 (GMT +
0)

date_default_timezone_set('UTC');
echo 'Date: ' . date('Y-m-d H:i:s'); // Hasil: 2017-01-17
14:36:12 (GMT + 0)
echo 'Timestamp: ' . time(); // Hasil: 1484663772 (GMT +
0)
```

Hasil:

```
Date: 2017-01-17 15:36:12
Timestamp: 1484663772

Date: 2017-01-17 14:36:12
Timestamp: 1484663772
```

Pada contoh diatas terlihat bahwa hasil dari fungsi `date()` berubah setelah kita ubah `timezone` menjadi UTC (GMT + 0), namun `timestamp` yang dihasilkan fungsi `time()` selalu sama yaitu: 1484663772 (GMT + 0). Fungsi `time()` ini bermanfaat untuk mendapatkan waktu kedepan atau kebelakang sejak waktu saat ini, misal:

```
echo'Waktu sekarang: '.date('Y-m-d H:i:s').'\n';  
echo'1 menit kedepan: '.date('Y-m-d  
H:i:s',time()+60).'\n';  
echo'1 jam kedepan: '.date('Y-m-d  
H:i:s',time()+(60*60)).'\n';  
echo'1 hari kedepan: '.date('Y-m-d  
H:i:s',time()+(60*60*24)).'\n';  
echo'7 hari kedepan: '.date('Y-m-d  
H:i:s',time()+(60*60*24*7)).'\n';
```

Hasil :

```
Waktu sekarang: 2017-01-17 15:48:34  
1 menit kedepan: 2017-01-17 15:49:34  
1 jam kedepan: 2017-01-17 16:48:34  
1 hari kedepan: 2017-01-18 15:48:34  
7 hari kedepan: 2017-01-24 15:48:34
```

Meskipun dapat mengkalkulasi waktu kedepan dan kebelakang, fungsi ini maksimal hanya dapat menghitung waktu dalam detik, tidak bisa menghitung 1 minggu kedepan atau 1 bulan kedepan. Untuk mengatasi hal tersebut, kita dapat menggunakan fungsi `strtotime()`.

## I. Fungsi `Mktime`

Fungsi `mktime()` (singkatan: make time) pada *PHP* juga digunakan untuk mendapatkan waktu tertentu dengan *format* `timestamp`. *Format* penulisannya adalah:

```
mktime(hour, minute, second, month, day, year)
```

Biasanya, kita membaca argumen dari kiri kekanan. Nah, untuk fungsi `mktime()` agar memudahkan memahami argumen yang digunakan, maka argumen dibaca dari kiri ke kanan (untuk argumen `time`: H:i:s) dan kanan ke kiri (untuk argumen `date`: Y-d-m). Semua

argumen bersifat opsional, nilai *defaultnya* adalah tanggal dan waktu hari ini:

```
mktime(date('H'), date('i'), date('s'), date('n'),  
date('j'), date('Y'));
```

Berikut ini akan menghasilkan nilai yang sama yaitu :Ketiganya akan menghasilkan timestamp yang sama, namun, sejak *PHP* versi 5.1, jika kita menjalankan fungsi `mktime()` tanpa argumen, akan muncul peringatan agar menggunakan fungsi `time()`. Semua argumen berbentuk *integer*. Fungsi `mktime()` ini mirip sekali dengan `time()`, biasa jadi karena namanya yang juga mirip. `mktime()` lebih fleksibel dari `time()` karena memiliki banyak argumen, namun keleluasaan manipulasi waktu hanya terbatas pada unit waktu, yaitu, jam, menit, detik, bulan, hari, dan tahun.

```
<?php  
echo strtotime('now');  
echo mktime();  
echo time();
```

## J. Fungsi Strtotime

Fungsi `strtotime()` (baca: *string to time*) pada *PHP* digunakan untuk menghasilkan waktu tertentu dengan *format* timestamp. *Format* penulisan fungsi `strtotime()` adalah sebagai berikut:

```
strtotime(string, timestamp);
```

*String* disini mencerminkan tanggal baik berbentuk karakter tanggal (Date atau *Time Format*) maupun kata atau kalimat (dalam bahasa Inggris) yang merujuk pada waktu tertentu (*Relative Format*). Misal:

Fungsi `strtotime()` paling fleksibel karena dapat menghitung waktu dengan acuan yang fleksibel misal: menghitung waktu satu bulan kedepan, hari terakhir suatu bulan, satu hari yang lalu, dll, dimana hal ini sulit dilakukan oleh kedua fungsi lainnya.

```
echo strtotime('now'); // Relative Format  
echo strtotime('2017-01-16'); // Date format  
echo strtotime('17:10:00'); // Time format
```

## K. Soal Latihan (Essay)

1. Jelaskan Perbedaan Built-In Function dan User-Defined Function adalah?

2. Buatlah Script Use-defined Function berikut

```
Selamat datang Agus  
Anda login sebagai Admin
```

3. Jelaskan fungsi dari date\_diff, dan berikanlah contohnya!

## L. Soal Latihan (Pilihan Ganda)

1. Berikut adalah script Built-in Function yaitu fungsi date :

```
1 <?php  
2 echo date("...");  
3
```

Lengkapi (...) di samping agar menampilkan hari ini

- a. D
- b. M
- c. Y
- d. ddd

2. Berikut adalah fungsi di *PHP* sebutkan jenis fungsi yang digunakan...

```
<?php  
echo date("l");  
?>
```

- a. Private
- b. *While*
- c. Built-in
- d. Use-defined

3. Script *PHP* function yang tepat untuk menghasilkan tampilan di bawah ini adalah...

```
Angka Genap :  
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

a.

```
<?php
function cetak(){
    for($i=1;$i<=50;$i++){
        if($i%2==0){
            echo"$i ";
        }
    }
    echo"Angka Genap : <br>";
    cetak();
?>
```

b.

```
<?php
function cetak(){
    for($i=10;$i<=50;$i++){
        if($i%2==1){
            echo"$i ";
        }
    }
    echo"Angka Genap : <br>";
    cetak();
?>
```

c.

```
<?php
function cetak(){
    for($i=1;$i<=50;$i){
        if($i%1==0){
            echo"$i ";
        }
    }
    echo"Angka Genap : <br>";
    cetak();
?>
```

d.

```
<?php
function cetak(){
    for($i=1;$i++){
        if($i%2==1){
            echo"$i ";
        }
    }
    echo"Angka Genap : <br>";
?>
```

4. Berikut yang bukan merupakan *script Function* untuk *String* adalah...
  - a. Ucfirst ( )
  - b. Strlen ( )
  - c. Mktime ( )
  - d. Echo ( )
  
5. Manakah yang bukan merupakan contoh *Built-In Function* pada pemrograman *PHP* ?
  - a. Date ( )
  - b. Printf ( )
  - c. Phpinfo ( )
  - d. *Errorlog* ( )

# BAB 8

## FORM DAN HYPERLINK

### A. Tujuan Pembelajaran

1. Mengenalkan kepada mahasiswa konsep form dan *hyperlink* pada *HTML*.
2. Mahasiswa dapat memahami pembuatan form dan *hyperlink*.
3. Mahasiswa dapat membuat program menggunakan form dan *hyperlink*.

### B. Form

Form merupakan salah satu elemen *HTML* yang digunakan untuk menerima *input* dari pengunjung. Melalui *form* tersebut, pengunjung dapat melakukan isian data yang kemudian dikirimkan ke *server*, *form* sebagai halaman untuk *user* bisa berinteraksi dengan *web*. Form terdiri dari banyak elemen input, seperti : *textfield*, *password field*, radio button, check box, submit button, reset button, select *list* (drop-down *list*), *textarea*. Form berfungsi untuk membuat sebuah halaman web interaktif dan dinamis.

```
<form>
...
Input elements
...
</form>
```

*Method* pengiriman data ke tujuan, yaitu :

*GET* : mengirim data pada *server* dengan cara meletakkan data pada bagian akhir *URL* yang ditunjuk. *POST* : mengirim datanya terpisah.

## C. Text Field

*Text field* merupakan elemen input yang memungkinkan pengunjung memasukkan *text* dalam satu baris. Lebar karakter *text* field secara *default* adalah 20 karakter. Perintah dasar :

```
<input type="text">
```

## D. Password Field

*Password field* digunakan untuk memasukkan satu baris *text* dengan *format* ciphertext/password. Karakter yang ditampilkan di password field disamarkan menjadi tanda bintang (\*) atau bulatan (●). Perintah dasar untuk password field :

```
<input type="password">
```

Contoh :

```
<!doctype html>
<html>
  <head> <title>Password Field</title></head>
  <body>
    <form>
      Password :<input type = "password" name = "pwd" />
    </form>
  </body>
</html>
```

## E. Radio Button

Radio button digunakan untuk pengunjung memilih satu pilihan data yang terbatas. Perintah dasar untuk radio button :

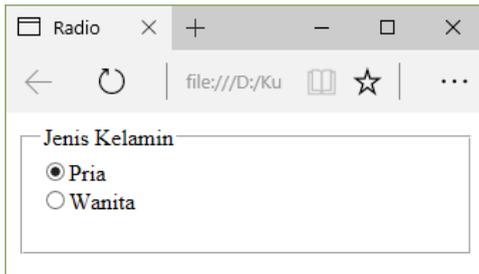
```
<input type="radio">
```

Contoh :

```
<html>
  <head> <title>radio</title></head>
  <body><fieldset><legend> Jenis Kelamin </legend>
    <form>
      <input type = "radio" name = "jk"
checked>pria<br>
      <input type = "radio" name = "jk">wanita<br>
    </form>
```

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

Hasilnya :

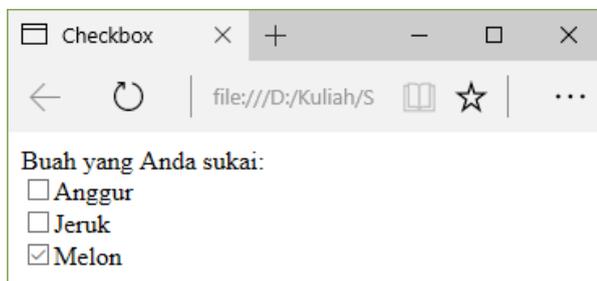


## F. Check Box

Check box digunakan untuk pengunjung dapat memilih beberapa pilihan data sekaligus atau lebih dari satu pilihan dari daftar pilihan yang ada. Perintah dasar dan contoh :

```
<html>
<head> <title>checked</title></head>
<body>
  <form> buah yang anda sukai: <br>
  <input type = "checkbox" name = "anggur">anggur<br>
  <input type = "checkbox" name = "jeruk">jeruk<br>
  <input type = "checkbox" name = "melon" checked
>melon<br>
  </form>
</body>
</html>
```

Hasilnya :



## G. Submit Button

Submit button digunakan untuk mengirimkan data dari halaman *web* ke *server*. Data yang dikirimkan ke halaman lain yang telah ditentukan pada atribut *action* di form. File yang ditentukan pada atribut *action* biasanya akan melakukan suatu aksi tertentu yang akan mengolah data yang dimasukkan dari elemen input. Perintah dasar :

```
<input type="submit">
```

Contoh :

```
<html>
<head> <title>Submit Button</title></head>
<body>
<form name="input" action="proses_input.php" method="get">
NIM : <input type="text" name="nim"/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

## H. Reset Button

Digunakan untuk membatalkan seluruh proses pengisian data yang dilakukan oleh elemen input form. Perintah dasar :

```
<input type="reset">
```

Contoh :

```
<html>
<head> <title>Reset Button</title></head>
<body>
<form name="input" action="proses_input.php"
method="get">
NIM : <input type="text" name="nim"/>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>
</body>
</html>
```

## I. Select List (Drop-down List)

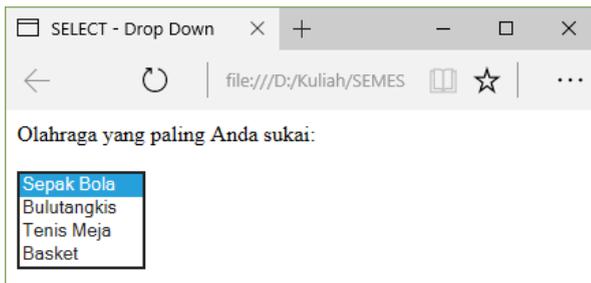
Select *list* digunakan untuk menampilkan daftar pilihan dalam bentuk drop-down *list*. Perintah dasar :

```
<select>
..
</select>
```

Contoh :

```
<html>
<head><title>select - drop down</title></head>
<body>
  <form>olahraga yang paling anda sukai:<br>
    <select name = "olahraga">
      <option value = "sepakbola" selected>sepak bola
      <option value = "bulutangkis">bulutangkis
      <option value = "tenismeja">tenis meja
      <option value = "basket">basket
    </select>
  </form>
</body>
</html>
```

Hasilnya :



## J. Textarea

*Textarea* digunakan sebagai input control form untuk memasukkan teks lebih dari satu baris. Elemen ini mempunyai atribut yaitu *name*, *cols*, dan *rows*. Atribut *name* mendefinisikan nama input control form element *textarea*. Atribut *cols* mendefinisikan jumlah kolom *textarea* yang terlihat, dan atribut *rows* mendefinisikan jumlah baris *textarea* yang

terlihat. Elemen yang harus berada di dalam *value* berisikan nilai elemen FORM. Perintah dasar :

```
<textarea>
name="name"
cols="number"
rows="number"
...
</textarea>
```

## K. Metode *GET* dan *POST*

*GET* dan *POST* merupakan *method* yang digunakan untuk mengirimkan data dari halaman client ke *server* melalui *webbrowser*. Masing-masing diterima oleh *server* dalam bentuk variable *array* *\$\_GET* dan *\$\_POST* yang bersifat *public*. Jika *method post* yang dipilih maka *variable* nilai yang dimasukkan di elemen input yang ada di form ketika dikirimkan ke *server* tidak akan ditampilkan di bagian *address bar browser*. Sedangkan jika *method get* yang digunakan, maka semua *variabel* nilai yang diisikan di dalam form ketika dikirim akan ditampilkan di bagian *address bar browser*. Pada contoh penggunaan form di atas (*form.html*), *method* yang digunakan adalah *method post*. Ketika nilai variabelnya dikirimkan ke *server* dan diproses, maka semua nilai variabel yang dimasukkan akan ditampilkan pada bagian *address bar browser*. Jika data yang akan dikirimkan ke *server* merupakan data yang sangat rahasia, sebaiknya *method post* yang digunakan.

### 1. Method *GET*

Fungsi *\$\_GET* merupakan fungsi yang digunakan untuk mengumpulkan nilai dari sebuah form yang dikirimkan menggunakan metode *GET*. Informasi yang dikirimkan dari sebuah form menggunakan metode *GET* dapat dilihat oleh semua orang, maksudnya adalah nilai dari form tersebut dapat dilihat dengan jelas pada jendela *Address bar* pada *browser*. Metode ini memiliki keterbatasan terhadap banyaknya informasi yang akan dikirimkan. Metode *GET* tidak cocok untuk nilai variabel yang penting dan besar, karena tidak dapat menyimpan lebih dari 2000 karakter. Contoh penggunaan metode *GET* (sumber : [www.w3schools.com](http://www.w3schools.com)) :



## M. Soal Latihan (Pilihan Ganda)

1. Atribut apakah yang ditambahkan di dalam form tombol submit agar dapat masuk ke halaman baru :
  - a. /form
  - b. Select
  - c. <a href>
  - d. Action
2. Method yang mengirimkan data atau nilai secara langsung menuju action adalah...
  - a. Get
  - b. Post
  - c. Send
  - d. Submit
3. Untuk membuat sebuah pilihan lebih dari satu di dalam form dapat digunakan input?
  - a. Radio Button
  - b. Select
  - c. Check box
  - d. *Text Field*
4. Button yang digunakan untuk membatalkan pengisian data atau nilai yang dilakukan oleh elemen input form adalah...
  - a. Get
  - b. Post
  - c. Send
  - d. Submit
5. Yang bukan merupakan elemen input dalam form adalah :
  - a. Radio Button
  - b. Check Box
  - c. Back Button
  - d. Submit Button

# BAB 9

## KONEKSI DATABASE

### A. Tujuan Pembelajaran

1. Mahasiswa dapat memahami langkah-langkah koneksi *PHP* dengan *MySQL*.
2. Mahasiswa dapat memahami perbedaan pengambilan *record* dari *database*.
3. Mahasiswa dapat membuat login dengan *session* dan *cookies*.

### B. *MySQL*

*MySQL* merupakan *Relational Database Management System* (RDBMS) yang diciptakan pada tahun 1979 oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database *engine* dengan indexing. *MySQL* merupakan salah satu jenis database yang terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan database sebagai sumber dan pengelolaan datanya.

Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia, (Sumber : <http://id.wikipedia.org/wiki/MySQL>). *MySQL* adalah database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (*PHP* dan Perl). Adapun keistimewaan dari *PHP* dan *MySQL* adalah keduanya merupakan bahasa pemrograman *web* dan RDBMS yang cross-platform (dapat dijalankan pada sistem operasi Linux dan Windows).

*PHP* terdiri dari aplikasi inti (*PHP core*), dan fungsi tambahan (*extension*). Dalam aplikasi inti *PHP*, *PHP* tidak menyediakan fungsi yang menangani *databaseMySQL*. Penanganan untuk fungsi database ini dipisahkan menjadi fungsi tambahan (*extension*). Selain *database*,

*extension* dalam *PHP* menyediakan berbagai fungsi lainnya untuk keperluan yang lebih khusus, seperti pembuatan gambar, kriptografi, penanganan email, pembuatan PDF, dan lain-lain. Pada *MySQL*, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah kolom dan baris, dimana setiap kolom berisi sekumpulan data yang memiliki tipe yang sejenis, dan baris merupakan sekumpulan data yang saling berkaitan dan membentuk informasi. Kolom biasanya juga disebut sebagai field dan informasi yang tersimpan dalam setiap baris disebut *record*.

Untuk dapat menggunakan dan mengakses *databaseMySQL*, *PHP* menyediakan 3 cara koneksi: menggunakan PDO (*PHP Data Objects*), *mysqli extension* dan *mysqlextension*.

### C. Tipe Data MySQL

Berikut ini tabel tipe data dalam *MySQL* yang dapat digunakan dalam *database* :

<b>Tipe data</b>	<b>Ukuran</b>	<b>Keterangan</b>
Tinyint	1 byte	Nilai <i>integer</i> yang sangat kecil
Smallint	2 bytes	Nilai <i>integer</i> yang kecil
Mediumint	3 bytes	<i>Integer</i> dengan nilai medium
Int	4 bytes	<i>Integer</i> dengan nilai standar
Bigint	8 bytes	<i>Integer</i> dengan nilai besar
Float	4 bytes	Bilangan decimal dengan <i>single-precision</i>
<i>Double</i>	8 bytes	Bilangan decimal dengan <i>double-precision</i>
Decimal (M, D)	M bytes (D+2, if M < D)	Bilangan float (decimal) yang dinyatakan sebagai <i>string</i>
Char (M)	M bytes, 1 <= M <= 255	<i>String</i> karakter dengan Panjang yang tetap
Varchar (M)	L+1 bytes, L <= M and 1 <= M <= 255	<i>String</i> karakter dengan Panjang yang tidak tetap
Tinyblob	L+1 bytes, L<2 <sup>8</sup>	BLOB (Binary Large <i>Object</i> ) yang sangat kecil
Blob	L+1 bytes, L<2 <sup>16</sup>	BLOB berukuran kecil
Mediumblob	L+1 bytes, L<2 <sup>24</sup>	BLOB berukuran sedang
Longblob	L+1 bytes, L<2 <sup>32</sup>	BLOB berukuran besar
Tinytext	L+1 bytes, L<2 <sup>8</sup>	<i>Stringtext</i> yang sangat kecil
Text	L+1 bytes, L<2 <sup>16</sup>	<i>String</i> teks berukuran kecil

Mediumtext	L+1 bytes, L<2 <sup>24</sup>	String teks berukuran medium
Longtext	L+1 bytes, L<2 <sup>32</sup>	String teks berukuran besar
Enum ('V1', 'V2',...)	1 or 2 bytes (65535 values max)	Enumerasi, kolom dapat diisi dengan 1 member enumerasi
Set ('Val1', 'Val2',...)	1, 2, 3, 4, or 8 bytes, (64 max)	Himpunan, kolom dapat diisi dengan beberapa nilai anggota himpunan.
Date	3 Bytes	"1000-01-01" sampai "9999-12-31"
Time	3 bytes	"-832:59:59" sampai "838:59:59"
Datetime	8 bytes	"1000-01-01 00:00:00" sampai "9999-12-31 23:59:59"
Timestamp	4 bytes	Range : 19700101000000 (suatu nilai tanggal pada tahun 2037)
Year	1 byte	1901 sampai 2155
Null		Nilai kosong (hampa)

## D. Koneksi PHP dan MySQL

PHP merupakan bahasa pemrograman yang selalu di *update* dan berkembang mengikuti teknologi terbaru. Saat ini, Pemrograman Berorientasi Objek (*Object Oriented Programming*) merupakan trend pemrograman PHP, dan hal ini juga mempengaruhi cara mengakses database MySQL dari PHP.

PHP memiliki 3 cara pengaksesan MySQL, yakni melalui PDO (*PHP Data Objects*), *mysqli extension* dan *mysqlextension*. PDO menggunakan pemrograman objek, *mysqli extension* tersedia dalam bentuk objek dan prosedural (diakses melalui fungsi-fungsi) sedangkan *mysqlextension* sepenuhnya menggunakan pemrograman prosedural.

### 1. Koneksi MySQL dengan *mysqlextension*

Untuk mengakses MySQL dari PHP, kita menggunakan fungsi-fungsi seperti *mysql\_connect()*, *mysql\_query()*, dan *mysql\_fetch\_array()*. Fungsi-fungsi ini tergabung ke dalam *mysqlextension* (saat itu PDO dan *mysqli extension* masih jarang digunakan). Extension di dalam PHP adalah istilah yang merujuk kepada kumpulan fungsi program tambahan yang membuat PHP dapat mendukung berbagai fitur baru. Namun sekarang (tepatnya mulai PHP versi 5.5.0) PHP memutuskan untuk membuat

*mysqlextension* berstatus deprecated. Yang artinya pengaksesan database *MySQL* menggunakan fungsi *mysqlextension* sudah tidak disarankan lagi.

## 2. Koneksi *MySQL* dengan *mysqli extension*

Sebagai pengganti *mysqlextension*, *PHP* menyediakan *mysqliextension* (*mysqli* merupakan singkatan dari *MySQL Improved*). *mysqliextension* ini pada dasarnya adalah perbaikan dari *mysqlextension* dan dikembangkan untuk mendukung fitur-fitur terbaru untuk *MySQL 4.1* keatas. Hampir semua fungsi yang ada pada *mysqlextension* juga tersedia pada *mysqli*. Syntax (aturan penulisan) *mysqli* sangat mirip dengan *mysqlextension*. Sehingga jika anda telah lama menggunakan *mysqlextension*, akan sangat mudah untuk beralih menggunakan *mysqli extension*.

Selain menggunakan *mysql* maupun *mysqli extension*, cara ketiga untuk pengaksesan database *MySQL* dari *PHP* adalah dengan menggunakan *PHP Data Objects* (atau sering disingkat dengan *PDO*).

## 3. Koneksi *MySQL* dengan *PDO (PHP Data Objects)*

*PDO (PHP Data Objects)*, adalah *extension* atau penambahan fitur dalam *PHP* yang dirancang sebagai interface universal untuk pengaksesan berbagai jenis database (tidak hanya *MySQL*). Contohnya, jika kita menggunakan *PDO* dalam menulis kode pemrograman, lalu suatu saat *website* kita bertukar database dari *MySQL* ke *Oracle*, maka kita tidak perlu mengubah semua kode program, cukup mengubah cara pemanggilan *PDO* diawal program saja.

Dari ketiga cara koneksi *PHP* dengan *MySQL* ini, metode yang disarankan adalah menggunakan *mysqli* atau *PDO*. Sebagai perbandingan fitur antara ketiga jenis koneksi *PHP-MySQL* ini, berikut adalah tabel perbandingan antara *mysqli*, *PDO* dan *mysqlextension* yang bersumber dari manual resmi *PHP*:

Comparison of MySQL API options for PHP			
	PHP's mysqli Extension	PDO (Using PDO MySQL Driver and MySQL Native Driver)	PHP's MySQL Extension
PHP version introduced	5.0	5.0	Prior to 3.0
Included with PHP 5.x	yes	yes	Yes
MySQL development status	Active development	Active development as of PHP 5.3	Maintenance only
Recommended by MySQL for new projects	Yes - preferred option	Yes	No
API supports Charsets	Yes	Yes	No
API supports server-side Prepared Statements	Yes	Yes	No
API supports client-side Prepared Statements	No	Yes	No
API supports Stored Procedures	Yes	Yes	No
API supports Multiple Statements	Yes	Most	No
Supports all MySQL 4.1+ functionality	Yes	Most	No

## E. PHP Data Objects

Untuk mengetahui secara sekilas perbedaan cara pengaksesan database *MySQL* menggunakan *mysqlextension*, *mysqli extension*, dan *PDO*.

## F. Langkah-langkah Koneksi *MySQL*

### 1. Koneksi *PHP* dengan *MySQL* (*mysql\_connect*)

Membuat koneksi antara *PHP* dengan *MySQL* menggunakan *mysqlextension*, *PHP* menyediakan fungsi *mysql\_connect* ( ). Fungsi ini digunakan untuk melakukan uji dan koneksi kepada server database *MySQL*. Fungsi *mysql\_connect* () membutuhkan 3 argumen, dan mengembalikan nilai fungsi berupa 'variabel koneksi' ke *MySQL*. Sintaks :

```
$link = mysql_connect ("host", "username", "password");
```

*\$link* merupakan variabel yang akan menampung hasil dari fungsi *mysql\_connect* (). Jika berhasil terhubung dengan *MySQL*, *\$link* akan berisi nilai yang berfungsi sebagai 'link koneksi' dengan *MySQL*. Link koneksi ini akan dibutuhkan sepanjang program *PHP-MySQL* nantinya. Di dalam *PHP*, variabel *\$link* ini dikenal dengan istilah handler, yakni

	variabel yang meng-handle koneksi ke <i>MySQL</i> .
<i>host</i>	adalah argumen pertama dari fungsi <i>mysql_connect()</i> . Nilai <i>mysql_host</i> diisi dengan alamat komputer dimana <i>MySQLServer</i> berjalan. Jika anda menjalankan <i>MySQL</i> menggunakan <i>XAMPP</i> di komputer yang sama dengan tempat <i>webserver</i> Apache berjalan, maka alamat ini bisa diisi dengan ' <i>localhost</i> ' atau ' <i>127.0.0.1</i> '. Namun jika <i>MySQLserver</i> dijalankan dari komputer lain, argumen ini akan berisi IP address dari komputer tersebut.
<i>username</i>	adalah nama user <i>MySQL</i> dimana kita akan login. Seperti ' <i>root</i> ', ' <i>admin</i> ', ' <i>andi</i> ' dan lain-lain tergantung kepada user <i>MySQL</i> yang telah terdaftar di <i>server</i> . Cara membuat dan menghapus user <i>MySQL</i> telah saya bahas pada Tutorial Cara Membuat dan Menghapus User <i>MySQL</i> ( <i>CREATE USER</i> ). Jika anda menggunakan <i>MySQL</i> bawaan <i>XAMPP</i> , user ' <i>root</i> ' dapat digunakan.
<i>password</i>	diisi dengan password dari user yang dibuat pada argumen <i>mysql_user</i> .

- Memilih *database* yang akan digunakan di *server* (*mysql\_select\_db()*)

Sintaks *mysql\_select\_db()* digunakan untuk melakukan koneksi kepada *database* pada *server* yang berhasil terkoneksi.

Sintaks :

```
$pilih = mysql_select_db("namadatabase", $link);
```

<i>\$pilih</i>	berisi status koneksi kepada database.
<i>\$link</i>	merupakan koneksi kepada <i>server</i> database yang berhasil.
<i>namadatabase</i>	adalah nama database yang akan dikenai proses.

- Menjalankan *queryMySQL* dari *PHP* (*mysql\_query()*)

Untuk menjalankan *queryMySQL*, *PHP* menyediakan fungsi *mysql\_query()*. Fungsi ini dijalankan setelah koneksi berhasil dilakukan dengan fungsi *mysql\_connect()*. Fungsi *mysql\_query()* merupakan fungsi yang akan sering kita gunakan, karena dengan fungsi inilah *PHP* akan melakukan 'percakapan' dengan *MySQL*. Fungsi *mysql\_query()* membutuhkan 2 argumen, dimana argumen pertama diisi dengan *queryMySQL*, dan argumen kedua diisi dengan *link* koneksi hasil fungsi *mysql\_connect()*. Argumen kedua ini bersifat opsional. Sintaks :

```
$result=mysql_query("query_mysql",[$link_koneksi_mysql]);
```

`$result` adalah variabel yang akan menampung hasil dari fungsi `mysql_query()`. Jika fungsi `mysql_query()` berhasil dijalankan, variabel `$result` akan berisi hasil *query*, namun jika *query* gagal, variabel ini akan berisi nilai *booleanFALSE*. Variabel `$result` bertipe *resources* (sama dengan hasil dari fungsi `mysql_connect()`). Nama dari variabel ini tidak harus ditulis sebagai `$result`, anda bebas jika ingin menggantinya dengan nama lain.

`"query_mysql"` adalah argumen pertama fungsi `mysql_query()`. Pada bagian inilah kita menulis *queryMySQL*. Query tersebut selanjutnya akan dikirim kepada *MySQLServer* untuk diproses. Argumen ini harus bertipe *string* dan mengikuti aturan penulisan *string* didalam *PHP*. Contoh *query* ini, misalnya "SELECT \* FROM mahasiswa", atau "CREATE DATABASE universitas".

`$link_koneksi_mysql` adalah argumen kedua dari fungsi `mysql_query()`. Argumen ini diisi dengan variabel hasil pemanggilan fungsi `mysql_connect()`. Argumen ini bersifat opsional, dan jika diabaikan *PHP* akan menggunakan variabel `mysql_connect()` yang sedang terkoneksi saat ini.

#### 4. Menampilkan Tabel *MySQL* dari *PHP* (`mysql_fetch_row`)

Perintah `SELECT` adalah *queryMySQL* yang paling sering digunakan. Query ini berfungsi untuk menampilkan data dari database. Data yang ditampilkan *MySQL* biasanya diberikan dalam bentuk tabel yang terdiri dari baris dan kolom.

Untuk menampilkan data *MySQL* di dalam *PHP*, hasil dari fungsi `mysql_query()` harus diproses lebih lanjut agar bisa ditampilkan di dalam *webbrowser*. *PHP* menyediakan banyak cara untuk menampilkan hasil *queryMySQL*. Salah satunya akan kita bahas dalam tutorial kali ini dengan fungsi: `mysql_fetch_row()`.

Fungsi `mysql_fetch_row()` digunakan untuk menampilkan tabel secara baris per baris. Fungsi `mysql_fetch_row()` akan mengembalikan nilai 1 baris dari sebuah tabel pada setiap

pemanggilan. Sehingga untuk dapat menampilkan seluruh isi tabel, `funcsimysql_fecth_row()` harus dipanggil secara berulang. Berikut adalah *format* dasar penulisan fungsi `mysql_fetch_row()` dalam *PHP* :

```
$row = mysql_fetch_row($result)
```

<code>\$row</code>	adalah variabel yang akan menampung hasil <code>funcsimysql_fetch_row()</code> . Hasil dari fungsi <code>mysql_fetch_row()</code> berupa tipe <code>dataarray</code> dengan <i>key</i> merujuk kepada urutan kolom, dan <i>value</i> -nya (nilai dari <i>array</i> ) adalah isi dari kolom tersebut.
<code>\$result</code>	adalah variabel inputan <i>query</i> untuk <code>funcsimysql_fecth_row()</code> . Variabel ini berasal dari fungsi <code>mysql_query()</code> .
<code>\$link_koneksi_mysql</code>	adalah argumen kedua dari <code>funcsimysql_query()</code> . Argumen ini diisi dengan variabel hasil pemanggilan fungsi <code>mysql_connect()</code> . Argumen ini bersifat opsional, dan jika diabaikan <i>PHP</i> akan menggunakan variabel <code>mysql_connect()</code> yang sedang terkoneksi saat ini.

Variabel `$row` hasil `funcsimysql_fetch_row()` bertipe *array*. Jika sebuah baris di dalam tabel *MySQL* terdiri dari 3 kolom, maka variabel `$row` akan berisi nilai sebagai berikut:

```
$row[0]=nilai_kolom_1,
$row[1]=nilai_kolom_2,
$row[2]=nilai_kolom_3.
```

Index *array* dimulai dari 0, dan karena itu kolom 1 berada pada index 0, bukan index 1. Yang perlu menjadi perhatian, `funcsimysql_fetch_row()` hanya membaca 1 baris pada sekali pemanggilan. Jika tabel kita terdiri dari 5 baris, maka fungsi `mysql_fecth_row()` harus diulang sebanyak 5 kali.

#### 5. Menampilkan Tabel *MySQL* dari *PHP* (`mysql_fetch_array`)

Fungsi `mysql_fetch_array()` pada dasarnya hampir sama dengan `funcsimysql_fetch_row()`, namun fungsi `mysql_fetch_array()` memberikan alternatif cara menampilkan data *MySQL* dengan lebih mudah.

Jika dalam fungsi `mysql_fetch_row()` hasil fungsinya berupa *array* dengan *key* atau indexnya berupa angka, seperti : `$row[1]`, `$row[2]` dan `$row[3]`, maka di dalam `mysql_fetch_array()`, kita bisa menggunakan nama kolom dari tabel *MySQL* sebagai *key* atau index *array* seperti berikut ini: `$row['nim']`, `$row['nama']` dan `$row['umur']`. Cara penulisan seperti ini akan memudahkan pembuatan program.

Selain menggunakan nama kolom pada index *array*, fungsi `mysql_fetch_array()` juga bisa menggunakan index penomoran kolom seperti `mysql_fetch_row()`, atau bahkan keduanya sekaligus. Untuk mengatur fitur ini, fungsi `mysql_fetch_array()` membutuhkan parameter tambahan. Berikut adalah *format* dasar penulisan fungsi `mysql_fetch_array()` dalam *PHP*:

```
$row= mysql_fetch_array($result, ['tipe_hasil'])
```

<code>\$row</code>	adalah variabel yang akan menampung hasil fungsi <code>mysql_fetch_array()</code> . Hasil dari fungsi <code>mysql_fetch_array()</code> berupa tipe data <i>array</i> dengan <i>key</i> yang bergantung kepada konstanta 'tipe_hasil'.
<code>\$result</code>	adalah variabel hasil menjalankan <code>queryMySQL</code> yang berasal dari fungsi <code>mysql_query()</code> .
<code>'tipe_hasil'</code>	merupakan konstanta opsional yang digunakan untuk mengatur index <i>array</i> dari hasil fungsi <code>mysql_fetch_array()</code> . Konstanta yang bisa digunakan adalah salah satu dari ketiga nilai berikut: <code>MYSQL_NUM</code> , <code>MYSQL_ASSOC</code> atau <code>MYSQL_BOTH</code> . Jika argumen ini tidak ditulis, nilai <i>default</i> nya adalah <code>MYSQL_BOTH</code> .

Jika argumen kedua dari fungsi `mysql_fetch_array()` diisi `MYSQL_NUM`, maka index *array* akan menjadi angka (sama seperti hasil `mysql_fetch_row()`), dan bisa ditampilkan seperti: `$row[1]`, `$row[2]` dan `$row[3]`. Jika argumen kedua diisi `MYSQL_ASSOC`, index *array* akan menjadi nama kolom, dan bisa ditampilkan seperti: `$row['nim']`, `$row['nama']` dan `$row['umur']`. Opsi ketiga, jika argumen diisi `MYSQL_BOTH` maka index *array* mendukung 2 pemanggilan, yakni berupa angka dan nama kolom.

Sama seperti fungsi `mysql_fetch_row()`, `mysql_fetch_row()` hanya membaca 1 baris pada sekali pemanggilan. Jika tabel kita terdiri dari 5 baris, maka fungsi `mysql_fetch_array()` harus diulang sebanyak 5 kali (atau menggunakan metoda perulangan `while` seperti yang kita lakukan pada tutorial `mysql_fetch_row()`).

#### 6. Menampilkan Pesan *ErrorMySQL* (`mysql_errno` dan `mysql_error`)

Agar lebih informatif, ketika *query* tidak berjalan dengan semestinya, *PHP* menyediakan 2 buah fungsi untuk menampilkan pesan *error* dari *MySQL*, yaitu fungsi `mysql_errno()` dan `mysql_error()`. Kedua fungsi ini membutuhkan 1 buah argumen yang bersifat opsional, yakni variabel hasil pemanggilan fungsi `mysql_connect()`. Tetapi, jika argumen tidak ditulis, *PHP* akan menggunakan koneksi *MySQL* terakhir yang tersedia.

Fungsi `mysql_errno()` akan menampilkan nomor atau kode *errorMySQL*, sedangkan fungsi `mysql_error()` akan menampilkan penjelasan *error* tersebut.

## G. Session

Session merupakan hal yang cukup penting dalam aplikasi berbasis *web*. Dengan session memungkinkan programmer menyimpan informasi user secara semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di *server*, jadi tidak bisa diakses secara langsung oleh client.

Dalam aplikasi berbasis *web*, session banyak digunakan sebagai autentikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus *log-in* terlebih dahulu. Dalam proses *log-in* antara lain akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman. Di halaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Dalam penanganan *session* terdapat beberapa proses yang perlu diperhatikan:

1. Proses pembuatan *session* yaitu untuk menciptakan *session*, karena *session* tidak tercipta secara otomatis :

```
<?php session_start(); ?>
```

`session_start()` menandakan *server* menginginkan untuk memulai sesi dengan user. Ketika kita memanggil `session_start()` maka:

- Jika *server* tidak pernah berinteraksi dengan *user*, maka akan dibuat *session* baru
- Jika *user* sudah pernah berinteraksi, maka *data session* akan di simpan di *associative array* `$_SESSION`
- Kita dapat menyimpan data di `$_SESSION` dan menerimanya kembali untuk digunakan lagi nantinya

2. Proses pemeriksaan *session*

- Memberi nilai pada variabel *session*

```
<?php
    session_start();
    $_SESSION['status']=1;
?>
```

- Membaca nilai dari variabel *session*

```
<?php
    session_start();
    echo "Status =" . $_SESSION['status'];
?>
```

3. Proses penghapusan *session*

- Menghapus sebuah variabel *session*

```
<?php
    session_start();
    if(isset($_SESSION['status']))
        unset($_SESSION['status']);
?>
```

- Menghapus semua *session* pada pengguna tertentu

```
<?php
    session_destroy();
?>
```

## H. Cookies

Cookies adalah tempat penyimpanan data pada hardisk lokal milik pengunjung dan digunakan oleh halaman *web* untuk mengingat sesuatu informasi yang pernah dikirimkan klien. Sekalipun cookie merupakan suatu mekanisme bagi *webserver* untuk menyimpan informasi ke klien, sebenarnya terdapat batasan. Sebuah klien hanya dapat memegang 300 cookie pada suatu saat dan sebuah *server* hanya bisa mengirim 20 cookie ke sebuah klien. Sebuah cookie hanya bisa berukuran sampai 4 kilobyte, sehingga total memory hardisk yang digunakan cookie hanya mencapai 1,2 megabyte.

Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti *username*, *password* dsb. Selain user bisa melihat informasi yang disimpan, *user* juga bisa *men-disable cookies* itu sendiri. Jika cookies *di-disable*, maka program yang memanfaatkan cookies tentunya tidak akan berjalan dengan baik.

Cookies sendiri biasanya dipakai dalam aplikasi *shopping cart*. Biasa digunakan untuk menyimpan sementara, produk-produk yang dipilih oleh pengunjung pada saat berbelanja.

### 1. Membuat cookie :

```
Setcookie(nama_cookie, nilai_cookie)
```

Nama\_cookie harus berupa *string* yang menyatakan nama cookie yang akan diciptakan dan nilai\_cookie menyatakan nilai yang akan disimpan pada nama cookie.

Membuat cookie yang berasal dari nilai form :

```
Setcookie(nama_cookie, $_POST["nama_var"])
```

Atau

```
$_COOKIE["nama"]
```

### 2. Mengakses Cookie

Untuk mengakses cookie, perlu memakai superglobal `$_COOKIE`. Digunakan untuk mengambil cookie nama.

```
Setcookie(nama_cookie, $_GET["nama_var"])
```

### 3. Menambahkan tanggal kedaluwarsa

Agar data tetap diingat walaupun *browser* berakhir, harus mengatur tanggal kedaluwarsa. Hal ini dapat dilakukan menggunakan `setcookie()`, yaitu dengan memberikan waktu kedaluwarsa pada argumen ketiga, sintaks yang digunakan adalah:

```
Setcookie (nama_cookie,  
          nilai_cookie,  
          waktu_kedaluwarsa);
```

Untuk memperoleh waktu sekarang, dapat digunakan perintah `time()` atau menggunakan 3600 untuk menyatakan jumlah detik dalam 1 jam.

### 4. Menghapus Cookie

Untuk menghapus cookie, gunakan `setcookie()` dengan menyebutkan nama cookie pada argumen pertama dan *string* kosong pada argumen kedua. Cara lain, argumen kedua tidak disebutkan. Perlu diketahui, penghapusan suatu cookie hanya berimplikasi pada `$_COOKIE` pada skrip lain.

## I. Soal Latihan (Essay)

1. Lengkapi Script pada gambar dibawah :

```
1  <?php
2      $db_host = "localhost";
3      $db_user = "DB";
4      $db_pass = "";
5      $db_name = "mahasiswa";
6      $koneksi = mysql_connect($...,$..., $...) or die("Tidak bisa terhubung ke Database!");
7      if(mysqli_connect_errno()){
8          echo 'Gagal melakukan koneksi ke Database:'.mysqli_connect_error ();
9      }else{
10         echo 'Koneksi berhasil ^_^'; echo "<br />";
11     }
12     mysql_select_db($..., $...); or die("Tidak ada Database yang dipilih!");
13
```

2. Jelaskan fungsi yang menghasilkan *array* asosiatif yang mana index-nya sesuai dengan nama kolom yang diseleksi!

## J. Soal Latihan (Pilihan Ganda)

1. Yang bukan merupakan fungsi untuk mengakses *MySQL* dari *PHP* adalah :
  - a. *mysql\_connect()*
  - b. *mysql\_query()*
  - c. *mysql\_error()*
  - d. *mysql\_fetch\_array()*
2. Fungsi yang digunakan untuk mengambil data dari *MySQL* dalam *PHP*, terdapat beberapa macam, yang bukan merupakan fungsi pengambilan data adalah :
  - a. *mysql\_fetch\_array()*
  - b. *mysql\_fetch\_assoc()*
  - c. *mysql\_connect()*
  - d. *mysql\_fetch\_object()*
3. Fungsi untuk meng-koneksikan database ke *PHP* dan untuk menampilkan pesan *error* dengan cara :
  - a. *Mysqli\_query*
  - b. *mysql\_query*
  - c. *Mysqli\_connect\_errno*
  - d. *Mysqli\_where*

4. Perbedaan *mysql\_fetch\_array* dan *mysql\_fetch\_assoc* adalah :
  - a. *mysql\_fetch\_array* menampilkan baris dan *value* sesuai urutan yang di definisikan pada *query* SQL, sedangkan *mysql\_fetch\_assoc* mengembalikan berturut-turut sebagai *array*
  - b. *mysql\_fetch\_array* mengembalikan berturut-turut sebagai *array*, sedangkan *mysql\_fetch\_assoc* menampilkan baris dan *value* sesuai urutan yang di definisikan pada *query* SQL
  - c. *mysql\_fetch\_array* mengambil baris sebagai *array* asosiatif, *array* numerik dan juga ditampilkan oleh kedua asosiatif &*array* numerik, sedangkan *mysql\_fetch\_assoc* mengembalikan berturut-turut sebagai *array*
  - d. *mysql\_fetch\_array* mengembalikan berturut-turut sebagai *array*, sedangkan *mysql\_fetch\_assoc* mengambil baris sebagai *array* asosiatif, *array* numerik dan juga ditampilkan oleh kedua asosiatif &*array* numerik
  
5. Dibawah ini merupakan tiga (3) cara koneksi *database*, kecuali :
  - a. PDO (*PHP Data Objects*)
  - b. *mysqli extension*
  - c. *mysql\_connect*
  - d. *mysqlextension*



# BAB 10

## KONSEP OOP

### A. Tujuan Pembelajaran

1. Mengenalkan *Object Oriented Programming* kepada mahasiswa
2. Mahasiswa dapat mengetahui perbedaan pemrograman berorientasi objek (OOP) dengan pemrograman terstruktur (*prosedural*) dalam bahasa PHP.
3. Mahasiswa dapat membuat aplikasi web menggunakan OOP.

### B. OOP

*Object Oriented Programming (OOP)* atau dalam bahasa Indonesia diartikan Pemrograman Berbasis Objek, adalah salah satu cara membuat program (*programming paradigm*) dengan memecah alur program menjadi modul-modul sederhana yang disebut dengan objek. Konsep *Object Oriented Programming (OOP)* / Pemrograman Berorientasi Objek) adalah sebuah paradigma pemrograman (cara membuat program) yang sangat populer. Konsep OOP ini banyak digunakan dalam bahasa pemrograman populer seperti C++ dan Java. Setiap objek akan memiliki data (atribut yang menjelaskan tentang objek) dan prosedur (*function*) yang dikenal dengan *method*. OOP bukan sekadar cara penulisan sintaks melainkan suatu paradigma/cara pandang dalam menganalisis permasalahan dalam pemrograman. Dalam OOP, setiap bagian dari program dianggap sebagai *object*. Sebuah *object* mewakili suatu bagian program yang hendak dirancang.

Beberapa konsep OOP dasar antara lain:

- a. Encapsulation (*Class* dan *Object*)
- b. Inheritance (pewarisan sifat)
- c. Polymorphism

### C. Perbedaan OOP dan Prosedural

OOP berbeda dengan prosedural (struktur) programming. OOP akan memecah program menjadi *object-object*, sedangkan prosedural programming memecah program menjadi fungsi-fungsi (prosedur). Penulisan OOP memungkinkan dibagi dalam tim, karena konsep objek untuk memisahkan program menjadi bagian-bagian yang berdiri sendiri akan memudahkan dalam membuat program. Penulisan program secara prosedural, yakni menulis program-program dari baris pertama sampai dengan baris terakhir secara berurutan.

### D. Class, Object, Property, dan Method

Bagian dasar dari pemrograman berorientasi objek sering disebut sebagai *object*. Cara mudah untuk memahami *object* yaitu melalui ilustrasi. Sebagai contoh, Mobil adalah sebuah *object*. Sebuah mobil memiliki beberapa property atau bagian-bagian di dalamnya seperti warna, mesin, roda, pintu, dsb. Sebuah mobil juga dapat melakukan sesuatu seperti bergerak, berhenti, menyalakan lampu, mengatur suhu, dsb.

*Object* berupa kata benda. *Object* misalnya orang, mobil, rumah, ponsel, dsb. Namun, tidak selamanya *object* merupakan benda nyata yang nampak. *Object* dapat berupa akun rekening bank, berkas di komputer, database, pesan email, dsb.

*Class* merupakan penjelasan atau deskripsi yang mewakili sebuah *object*. Di dalam class terdapat penjelasan tentang suatu *object* berupa property yang dimiliki *object* tersebut, dan method yang dapat dikerjakan oleh *object* tersebut. Contohnya class Mobil. Dalam class Mobil terdapat penjelasan berupa property tentang merk, warna, tipe, kapasitas mesin mobil, dsb. Di dalam class tersebut juga terdapat method-method yang dapat dikerjakan oleh mobil seperti menyalakan mesin, maju, mundur, berbelok, transmisi, menyalakan AC, memutar musik, dsb.

Di antara keuntungan konsep OOP adalah adanya pengkapsulan (encapsulation) program dalam *Class* atau *object*. Programmer yang hendak menggunakan *Class* tidak perlu mengetahui isi dan alur kerja *Class* tersebut secara detail, akan tetapi cukup mengetahui bagaimana

cara menggunakannya. Sama halnya dengan orang yang hendak mengemudikan mobil. Orang tersebut tidak perlu mengetahui bagian-bagian mobil dan perangkat-perangkatnya secara menyeluruh. Dia tidak perlu mengetahui bagaimana mobil bisa menyala, bagaimana transmisi bekerja, atau bagaimana mesin mobil mampu menggerakkan roda. Yang perlu ia tahu hanya bagaimana cara menyalakan mesin, menjalankan mobil, menghentikan mobil, dan fungsi-fungsi lainnya.

Setiap *class* memiliki *property* atau sering disebut sebagai *attribute*. Property dari sebuah mobil misalnya merk, warna, tipe, kapasitas mesin mobil, dsb. Property dinyatakan dalam bentuk *variable* misalnya \$merk, \$warna, \$tipe, dsb.

*Method* merupakan sesuatu yang dapat dikerjakan oleh class. Dalam *PHP*, method didefinisikan menggunakan *function*. Method yang ada dalam class Mobil misalnya adalah method untuk menyalakan mesin, maju, mundur, berbelok, transmisi, menyalakan AC, memutar musik, dsb.

Aturan penamaan property dan method sebenarnya sama dengan *variable* atau fungsi. Akan tetapi, berdasarkan konvensi yang umum digunakan adalah penamaan berbasis *Mixed Caps/Lower CamelCaps*. Setiap kata selain kata pertama diawali dengan huruf kapital, dan masing-masing kata tidak diberi pemisah baik spasi maupun *underscore* (\_). Contoh penulisannya untuk *method*: mobilMaju, bukaKunciTangki. Sedangkan untuk property: \$kapasitasMesin, \$jenisTransmisi. Bentuk umum untuk mendefinisikan class, sebagai berikut :

```
class NamaClass{
    Deklarasi dan definisikan property di sini
    Deklarasikan semua method di sini
}
```

Penamaan class pada dasarnya sama dengan *variable*. Akan tetapi, dikecualikan untuk *stdClass*. Nama class *stdClass* sudah digunakan oleh *PHP* sebagai nama *class built-in*. Isi class terletak antara kurung kurawal buka ({} dan kurung kurawal tutup (}). Isi *class* berupa property maupun *method class* tersebut.

Cara mendefinisikan *method* sama halnya dengan fungsi biasa. Aturan penamaan *method* sama halnya dengan penamaan *variable*. Akan tetapi, jangan menggunakan penamaan yang diawali *double underscore* (\_\_) yaitu *\_\_construct()*, *\_\_destruct()*, dan *\_\_clone()*. Ketiga fungsi

tersebut memiliki arti tersendiri dalam *PHP*.

## E. Variabel `$this`

Variabel `$this` adalah sebuah variabel khusus dalam *OOPPHP* yang digunakan sebagai penunjuk kepada objek, ketika diakses dari dalam *class*. Dalam manual *PHP*, `$this` disebut dengan istilah: *pseudo-variable*.

## F. Enkapsulasi

Enkapsulasi atau pembungkusan adalah sebuah metoda untuk mengatur struktur *class* dengan tujuan menentukan hak akses pada *property* atau *method*. Untuk membatasi hak akses di dalam sebuah *class*, *Objek Oriented Programming* menyediakan 3 kata kunci, yakni *Public*, *Protected* dan *Private*. *Public* : maka seluruh kode program di luar *class* bisa mengaksesnya. *Protected* : *property* atau *method* tersebut tidak bisa diakses dari luar *class*, namun bisa diakses oleh *class* itu sendiri atau turunan *class* tersebut. *Private* : satu-satunya yang bisa mengakses adalah *class* itu sendiri. *Class* lain tidak bisa mengaksesnya, termasuk *class* turunan.

## G. Inheritance (pewarisan atau penurunan)

Inheritance adalah konsep pemrograman dimana sebuah *class* dapat menurunkan *property* dan *method* yang dimilikinya kepada *class* lain, dg kata kunci “extends”.

## H. Constructor

Constructor (bahasa indonesia: konstruktor) adalah *method* khusus yang akan dijalankan secara otomatis pada saat sebuah objek dibuat (instansiasi), yakni ketika perintah “new” dijalankan. Penulisan *method* Constructor dan Destructor di *PHP* didahului dua *underscore* (`__construct()` dan `__destruct()`). Kedua *method* ini merupakan *magic method/method* ajaib dalam bahasa *PHP*. Istilah *instantiate* sendiri berarti membuat sebuah *object* dari sebuah *class*.

## I. Destructor

Destructor (bahasa indonesia: destruktur) adalah method khusus yang dijalankan secara otomatis pada saat sebuah objek dihapus. Di dalam *PHP*, seluruh objek secara otomatis dihapus ketika halaman *PHP* dimana objek itu berada selesai diproses. Dalam *PHP*, destructor dibuat menggunakan method : `__destruct()`.

## J. Soal Latihan (Essay)

1. Jelaskan tentang konsep OOP, dan berikan contoh!
2. Apakah definisi tentang Method dalam OOP?
3. Salin dan Lengkapi kekurangan Script di bawah ini agar tampilan Seperti di bawah ini :

Makul Web pro

```
<?php
class webpro{
    //method
    function makul(){
        return "Makul Web pro <br/>";
    }
}
//instansiasi
$. . . . = new . . . ();

//memanggil method
echo $kuliah->. . . ();
```

## K. Soal Latihan (Pilihan Ganda)

1. Konsep OOP yang digunakan untuk mengencapsulasi / membungkus data dan abstraksi prosedural yang diperlukan dalam menggambarkan isi dan tingkah laku berbagai aktifitas tersebut?
  - a. Class
  - b. Method
  - c. Property
  - d. Constuctor

2. Encapsulasi data yang digunakan sebagai kunci, dengan menggunakan...
  - a. `$this`
  - b. `Return`
  - c. `Class kuliah`
  - d. `Webpro`
  
3. Data yang bisa diubah dalam sebuah konsep OOP adalah ...
  - a. `Method`
  - b. `Attribut`
  - c. `Properties`
  - d. `Value`
  
4. Untuk membatasi hak akses di dalam sebuah class, OOP menyediakan 3 kata kunci, kecuali...
  - a. `Public`
  - b. `Protected`
  - c. `Properties`
  - d. `Private`
  
5. Mobil merupakan salah satu kata benda, di dalam OOP kata benda termasuk dalam ...
  - a. `Object`
  - b. `Method`
  - c. `Class`
  - d. `Property`

# BAB 11

## CSS

### A. Tujuan Pembelajaran

1. Mengenalkan *CSS* kepada mahasiswa
2. Mahasiswa dapat mengetahui sintaks penulisan *CSS* yang *diembedded* dalam bahasa pemrograman *PHP*.
3. Mahasiswa dapat membuat aplikasi *web* dengan tampilan menggunakan *CSS*

### B. CSS

*WebLayout* merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Salah satu *web* layout yang sering digunakan adalah *CSS* (*Cascading Style Sheets*). *CSS* adalah bahasa penulisan yang digunakan untuk mendeskripsikan penampilan sebuah dokumen *markup*. *CSS* bukanlah merupakan bahasa pemograman, karena hanya mengatur tampilan dokumen saja, misal : ukuran, warna, spasi, margin, dll.

*HTML* dirancang tidak ditujukan untuk mendesain sebuah halaman *web*, tapi hanya untuk menampilkan konten saja. Pada saat *HTML* beranjak ke versi 3.2, diperkenalkan *tag* `<font>` yang digunakan untuk mengatur tampilan sebuah teks, seperti jenis font yang digunakan, warna, dan ukuran. Tapi dengan adanya *tag* ini, malah jadi menyulitkan proses development, karena *tag* ini mesti disematkan di banyak halaman untuk mendapatkan hasil style yang sama. Untuk itu, dibuatlah *CSS*, yang memisahkan style dari halaman *web*, sehingga antara konten pada *HTML* dan desain tampilan pada dokumen *CSS* dapat dikerjakan di dua tempat berbeda.

## C. Metode Penerapan CSS

Metode penerapan CSS terdapat beberapa cara, yaitu :

### 1. *Inline style sheet*

CSS didefinisikan langsung pada *tagHTML* yang bersangkutan. Cara penulisannya cukup dengan menambahkan atribut `style="..."` dalam *tagHTML* tersebut. Style hanya akan berlaku pada *tag* yang bersangkutan, dan tidak akan memengaruhi *tagHTML* yang lain. Contoh :

```
<h1><font style= "text-align:center;
color:lightsteelblue;"> Selamat Datang,
Programmer!</h1>
<p>Selamat belajar pemrograman web</p>
```

### 2. *Embedded Style Sheet (Internal)*

CSS didefinisikan terlebih dahulu dalam `tag<style> ... </style>` di atas `tag<body>`. Pada pendefinisian ini disebutkan atribut-atribut CSS yang akan digunakan untuk *tag-tagHTML*, yang selanjutnya dapat digunakan oleh *tagHTML* yang bersangkutan.

```
<style type="text/css">
  h1 { text-align: center; color: lightsteelblue;
}
</style>

<h1>Selamat Datang, Programmer!</h1>
<p>Selamat belajar pemrograman web</p>
```

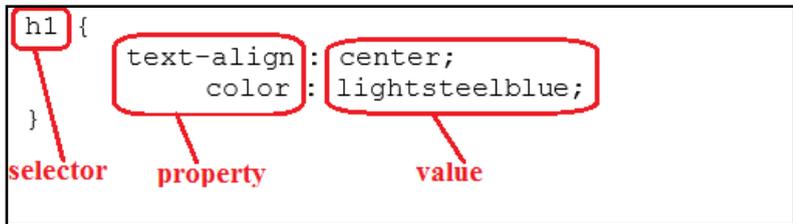
### 3. *Embedded Style Sheet (External)*

CSS didefinisikan terlebih dalam sebuah *file* (\*.css). Pada pendefinisian ini disebutkan atribut-atribut CSS yang akan digunakan untuk *tag-tagHTML*, yang selanjutnya nama *file* di panggil dengan syntax *html* pada *fileHTML*.

```
<h1>Selamat Datang, Programmer!</h1>
<p>Selamat belajar pemrograman web</p>

h1 {
  text-align : center;
  color : lightsteelblue;
}
```

## SintaksCSS



Selector adalah bagian *CSS* untuk merujuk elemen *HTML* yang ingin dikenakan style. Property dan *value* *CSS* dipisahkan dengan tanda titik dua (:). Pada *CSS* dapat diberikan lebih dari satu property style dengan memisahkan antar property dengan tanda titik koma (;). property dan *value* disimpan di dalam kurung kurawal { dan }. Beberapa jenis selector :

- ✓ Selector Element : mengacu elemen *HTML* berdasarkan nama elemennya.
- ✓ Selector class : mengacu elemen berdasarkan nama classnya. Nama class pada elemen diacu dengan nilai atribut class diawali oleh tanda titik (.)
- ✓ Selector id : mengacu elemen berdasarkan nama idnya. Nama id pada elemen diacu dengan nilai atribut id diawali oleh tanda pagar (#).

### D. Soal Latihan (Essay)

1. Buatlah sebuah *Script Form Login* menggunakan *CSS* yang terdiri dari Username, Password dan inputan Login!
2. Sebutkan dan jelaskan 3 cara penulisan *CSS* di *HTML*
3. Tuliskan cara pemanggilan *file* style *css* yang benar !

### E. Soal Latihan (Pilihan Ganda)

1. Pada *style* di *CSS* terdapat berbagai macam atribut untuk mengatur halaman *HTML* yang kita hubungkan dengan *style* *CSS* yang ada. Manakah yang merupakan atribut untuk mengatur warna huruf / *font* ?
  - a. bgcolor : red
  - b. fontcolor : red

- c. `color : red`
  - d. `colorfont : red`
2. `<h2 style="color:red; font-family: sans;">Ini judul artikel</h2>`  
Script di atas merupakan potongan dari ...
- a. Internal CSS
  - b. External CSS
  - c. Inline CSS
  - d. Dalam CSS
3. Salah satu jenis selektor CSS adalah *class*. Untuk mendeklarasikan sebuah selektor *class* tanda yang digunakan di depan nama *class* adalah ...
- a. Tanda “:”, (titik dua)
  - b. Tanda “.”, (titik)
  - c. Tanda “#”, (kres)
  - d. Tanda “;”, (titik koma)
4. Dalam CSS dikenal dengan konsep box model yang menyatakan bahwa setiap elemen blok *HTML* memiliki area padding, margin, dan border. Urutan yang benar dari letak ketiga area tersebut dalam suatu elemen *HTML* adalah...
- a. Padding-border-margin
  - b. Margin-padding-border
  - c. Border-margin-padding
  - d. Padding-margin-border

# BAB 12

## Javascript

### A. Tujuan Pembelajaran

1. Mengenalkan javascript kepada mahasiswa
2. Mahasiswa dapat mengetahui sintaks penulisan javascript yang *diembedded* dalam bahasa pemrograman *PHP*.
3. Mahasiswa dapat membuat operasi javascript dengan *PHP* database *MySQL*

### B. Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk menambahkan fitur interaktif pada *website* yang berjalan di sisi *Client*. Program javascript dapat mengubah isi elemen *web* yang sedang ditampilkan tanpa adanya perubahan pada alamat *URL*. JavaScript seringkali disertakan dalam *fileHTML* atau *link* dari *fileHTML* dan dijalankan secara lokal oleh *webbrowser*. Ini berarti bahwa *server* bebas untuk mengerjakan sesuatu yang lain daripada pemrosesan instruksi untuk setiap klien. Hal ini telah membuat JavaScript pilihan yang lebih populer daripada bahasa yang memerlukan *server* untuk melakukan pengolahan.

Pemrograman javascript setidaknya yang diperlukan yaitu *browser* dan teks editor. Kode Program Java Script dapat dituliskan langsung pada *fileHTML*. Dengan menggunakan *tag* kontainer `<SCRIPT>`. Dan tidak perlu menuliskan program java script pada *file* terpisah.

*Tag* kontainer adalah *tag* yang diawali dengan `<NAMA_TAG>` dan diakhiri `</NAMA_TAG>`.

Contoh :

```
<HTML></HTML>
<BODY></BODY>
```

```
<HEAD></HEAD>
```

Tag kontainer `<SCRIPT>` mempunyai dua atribut tetapi hanya satu atribut `language` dengan "JavaScript" yang diisikan untuk memberitahukan kepada *browser* bahwa yang akan ditulis adalah JavaScript.

Contoh :

```
<SCRIPT LANGUAGE="JavaScript">  
  
    // program dimasukkan disini  
</SCRIPT>
```

Event handler adalah program yang bereaksi ketika terjadi event tertentu, event handler yang digunakan adalah "OnClick" yang memberitahukan ke komputer apa yang harus dilakukan jika event ini terjadi. Operator Java Script dibagi menjadi 6 yaitu : Aritmetik, Pemberian nilai (Assign), Pemanipulasian bit (bitwise), Pembanding, Logika, *String*. Tipe objek pemasukan data di javascript yaitu objek teks, objek checkbox, objek radio, objek *textarea*, objek *select*.

### C. JQuery

JQuery adalah pustaka JavaScript lintas-platform yang didesain untuk menyederhanakan *client-side scripting* pada HTML. Query pada intinya merupakan pustaka untuk memanipulasi DOM (*Document Object Model*). Dom adalah struktur-pohon representasi dari semua elemen yang terdapat dalam sebuah halamanWeb, dan jQuery menyederhanakan sintaks untuk mencari, menyeleksi, dan memanipulasi elemen-elemen DOM tersebut.

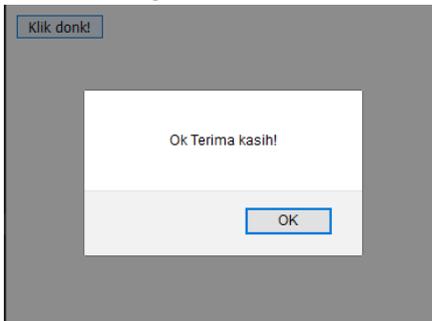
Sintaks pada jQuery didesain untuk memudahkan dalam navigasi sebuah dokumen, pemilihan elemen DOM, pembuatan animasi, penanganan event, dan pengembangan aplikasi berbasis Ajax. AJAX, singkatan dari "Asynchronous JavaScript and XML", merupakan metode suatu laman *web* menggunakan JavaScript untuk mengirim dan menerima data dari *server* tanpa harus menyegarkan (*refresh*) laman itu.

## D. Soal Latihan (Essay)

1. Sebutkan Kelebihan Javascript!
2. Jelaskan kegunaan dari penggunaan javascript di *PHP*! Dan berikan contoh!

## E. Soal Latihan (Pilihan Ganda)

1. Bagaimana penulisan dalam *HTML* untuk meletakkan script Javascript ?
  - a. `<script>`
  - b. `<javascript>`
  - c. `<js>`
  - d. `<jscript>`
2. Perhatikan gambar di bawah ini :



- Type javascript yang terdapat pada gambar di atas termasuk ...
- a. Alert
  - b. Danger
  - c. Announcement
  - d. Notice
3. Dalam javascript bisa di sisiapkan melau *file* eksternal dengan cara...
    - a. `<script src="file-eksternal.js"></script>`  
`<script>`
    - b. `document.write("Javascript itu keren!");`  
`</script>`
    - c. `javascript:alert("Nah! ini Javascript")`

- d. `<button onclick="alert('Ok Terima kasih!')">Klik donk!</button><script>`
4. Argumen dari fungsi Javascript *getElementById* adalah:
- a. nama suatu elemen *HTML* yang dibuat menggunakan atribut *ID*
  - b. nama suatu elemen *HTML* yang dibuat menggunakan atribut *class*
  - c. identitas suatu elemen yang dibuat menggunakan atribut *class*
  - d. identitas suatu elemen yang dibuat menggunakan atribut *ID*
5. Jenis-jenis node di Javascript adalah...
- a. Elemen, atribut, *text*, document
  - b. *HTML*, atribut, document
  - c. *CSS*, elemen, *text*, document
  - d. *HTML*, *CSS*, atribut, document

## DAFTAR PUSTAKA

- Muhammad, R. A, 2011, *Pemrograman Web Dinamis menggunakan PHP dan MySQL*, Andi Offset Yogyakarta.
- Pratama, A, 2018, *Belajar HTML Dasar Part 5: Pengertian Tag, Elemen, dan Atribut pada HTML*, Dunia Ilmu Komputer, dilihat 1 Agustus 2019, <https://www.duniailkom.com/belajar-html-pengertian-tag> .
- Pratama, A, 2019, *Tutorial Belajar PHP Part 5: Cara Menjalankan File PHP dengan XAMPP*, Dunia Ilmu Komputer, dilihat 2 Agustus 2019, <https://www.duniailkom.com/cara-menjalankan-file-php-dengan-xampp/> .
- Pratama, A, 2019, *Tutorial Belajar PHP Part 9: Aturan Dasar Penulisan Kode PHP*, Dunia Ilmu Komputer, dilihat 2 Agustus 2019, <https://www.duniailkom.com/aturan-dasar-penulisan-kode> .
- Pratama, A, 2019, *Tutorial Belajar PHP Part 10: Cara Penulisan Komentar dalam Kode PHP*, Dunia Ilmu Komputer, dilihat 2 Agustus 2019, <https://www.duniailkom.com/cara-penulisan-komentar-dalam-kode> .
- Pratama, A, 2019, *Tutorial Belajar PHP Part 11: Pengertian Variabel dan Cara Penulisan Variabel PHP*, Dunia Ilmu Komputer, dilihat 2 Agustus 2019, <https://www.duniailkom.com/pengertian-variabel>
- Pratama, A, 2014, *Tutorial Belajar PHP Part 13: Mengenal Tipe Data Integer dan Cara Penulisan Integer dalam PHP*, Dunia Ilmu Komputer, dilihat 3 Agustus 2019, <https://www.duniailkom.com/mengenal-tipe-data> .
- Pratama, A, 2014, *Tutorial Belajar PHP Part 14: Mengenal Tipe Data Float dan Cara Penulisan Float dalam PHP*, Dunia Ilmu Komputer, dilihat 3 Agustus 2019, <https://www.duniailkom.com/mengenal-tipe-data> .
- Pratama, A, 2014, *Tutorial Belajar PHP Part 15: Mengenal Tipe Data String dan Cara Penulisan String dalam PHP*, Dunia Ilmu Komputer, dilihat 3 Agustus 2019, <https://www.duniailkom.com/mengenal-tipe-data> .
- Pratama, A, 2014, *Tutorial Belajar PHP Part 16: Mengenal Tipe Data Boolean dan Cara Penulisan Boolean dalam PHP*, Dunia Ilmu

Komputer, dilihat 3 Agustus 2019,  
<https://www.duniaikom.com/mengenal-tipe-data> .

Pratama, A, 2014, Tutorial Belajar *PHP* Part 29: Pengertian dan Cara Penulisan Struktur Logika IF dalam *PHP*, Dunia Ilmu Komputer, dilihat 4 Agustus 2019, <https://www.duniaikom.com/tutorial-belajar-php-pengertian-dan-cara-penulisan-struktur> .

Mulyadi, M, 2019, Mencetak Beragam Pola dengan Konsep Perulangan, Codepolitan, dilihat 5 Agustus 2019,  
<https://www.codepolitan.com/mencetak-beragam-pola> .

Muhardian, A, 2015, Belajar *PHP*: Memahami 4 Jenis Perulangan dalam Pemrograman *PHP*, Petani Kode, dilihat 5 Agustus 2019,  
<https://www.petanikode.com/php-perulangan/> .

Munandar, A, 2016, Struktur Logika Perulangan Pada *PHP*, Bahasa Web, dilihat 5 Agustus 2019, <https://bahasaweb.com/perulangan-pada-php/> .

## GLOSARIUM

Atribut	: Merupakan informasi tambahan yang diberikan kepada <i>tag</i> . Informasi ini bisa berupa instruksi untuk warna dari <i>text</i> , besar huruf dari <i>text</i>
Array	: Merupakan sekumpulan struktur data yang dapat menyimpan banyak nilai ataupun elemen dalam sebuah variabel secara berurutan
Binary	: Sistem bilangan biner atau sistem bilangan basis dua adalah sebuah sistem penulisan angka dengan menggunakan dua simbol yaitu 0 dan 1. Sistem bilangan biner modern ditemukan oleh Gottfried Wilhelm Leibniz pada abad ke-17
Break	: dalam program akan menyebabkan program menghentikan perulangan (langsung keluar dari perulangan).
Boolean	: Bilangan dengan hasil <i>TRUE</i> dan <i>FALSE</i>
CSS	: Cascading Style Sheet merupakan aturan untuk mengatur beberapa komponen dalam sebuah <i>web</i> sehingga akan lebih terstruktur dan seragam
Define	: <i>define</i> digunakan untuk mendefinisikan sebuah variable (pin input - output dll) dengan nama variable baru, fungsi mendefinisikan variable ini agar dalam pembuatan program kita lebih mudah mengingat nama variabelnya
Double	: Jenis data pecahan dan desimal
Do-while	: pengecekan kondisi dilakukan di akhir perulangan, sehingga walaupun kondisi <i>FALSE</i> , perulangan akan tetap berjalan minimal 1 kali.
Echo	: menampilkan data dalam standard output
ELSE-IF	: bisa dibuat kode program yang akan menyeleksi berbagai kemungkinan yang bisa terjadi
HTTP	: <i>Hypertext Transfer Protocol</i> adalah protokol

- aplikasi untuk sistem informasi hypermedia yang didistribusikan, kolaboratif, dan kolaboratif
- HTTPS : *Hypertext Transfer Protocol Secure* adalah ekstensi dari *Hypertext Transfer Protocol*. Ini digunakan untuk komunikasi aman melalui jaringan komputer, dan banyak digunakan di Internet. Dalam HTTPS, protokol komunikasi dienkripsi menggunakan *Transport Layer Security* atau, sebelumnya, *Secure Sockets Layer*
- HTML : *Hypertext Markup Language*, salah satu yang memerintahkan bahasa pemrograman *web* desain juga biasa disebut *script* untuk menyusun dokumen-dokumen *web*
- Hyperlink : digunakan untuk menghubungkan beberapa halaman *web*
- IF : dalam bahasa pemrograman adalah sebuah struktur logika untuk membuat percabangan alur program
- IIS : *Internet Information Services* atau *Internet Information Server* adalah sebuah *HTTP webserver* yang digunakan dalam sistem operasi *server*
- Integer : Jenis data bilangan bulat, maksimum 32 bit
- JavaScript : bahasa pemrograman tingkat tinggi dan dinamis. JavaScript populer di internet dan dapat bekerja di sebagian besar penjelajah *web* populer seperti Google Chrome, Internet Explorer, Mozilla Firefox, Netscape dan Opera
- Konstanta : merupakan *variable* dengan nilai tetap. Konstanta diberikan nilai awal program dan nilainya tidak akan berubah selama program berjalan
- LAMP : istilah yang merupakan singkatan dari Linux, Apache, *MySQL* dan *Perl/PHP/Python*. Merupakan sebuah paket perangkat lunak bebas yang digunakan untuk menjalankan sebuah aplikasi secara lengkap.

- Lighttpd : *webserver* free, open-source, serta berkecepatan tinggi yang dirancang khusus untuk lingkungan yang kritis akan kecepatan
- Litespeed : sebuah terobosan baru dari teknologi *webserver* yang merupakan pengganti ApacheWebServer
- MAMP : singkatan dari Macintosh, Apache, Mysql dan Php
- MySQL : sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia
- Nginx : *server* HTTP dan Proxy dengan kode sumber terbuka yang bisa juga berfungsi sebagai proxy IMAP/POP3
- Notepad++ : sebuah penyunting teks dan penyunting kode sumber yang berjalan di sistem operasi Windows
- Oracle : relational database management system (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi
- PHP : *PHP* : Hypertext Presprocessor". *PHP* adalah bahasa pemrograman *web* bersifat *server-side* scripting yang menyatu dengan *HTML* untuk membuat halaman*web* dinamis
- PostgreSQL : sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian lisensi BSD. Peranti lunak ini merupakan salah satu basis data yang paling banyak digunakan saat ini, selain *MySQL* dan Oracle
- Printf : fungsi yang digunakan untuk mengatur *format* data
- Script : bahasa pemrograman yang menyediakan fasilitas penerjemahan serta kompilasi kode dalam satu rangkaian proses secara integratif sehingga memungkinkan kode dibuat dapat langsung dijalankan sebagai program secara dinamis
- Server HTTP : *serverweb* yang dapat dijalankan di banyak sistem

Apache	: operasi yang berguna untuk melayani dan memfungsikan situs <i>web</i> . Protokol yang digunakan untuk melayani fasilitas <i>web/www</i> ini menggunakan HTTP
<i>String</i>	: Jenis data teks / untaian karakter, kalimat
Sublime Text	: sebuah teks editor yang sangat canggih untuk coding, <i>markup</i> dan lain-lain
switch	: sebuah stuktur percabangan yang akan memeriksa suatu variabel, lalu menjalankan perintah-perintah yang sesuai dengan kondisi yang mungkin terjadi untuk variabel tersebut
Sybase	: sebuah perusahaan perangkat lunak komputer yang mengembangkan dan menjual sistem manajemen database (DBMS) dan produk middleware
Tomcat	: <i>webserver</i> yang juga dikembangkan oleh ApacheSoftware Foundation, dan karena itulah nama resminya adalah Apache Tomcat
WAMP	: singkatan dari Windows, Apache, <i>MySQL</i> , dan <i>PHP</i> . WAMP digunakan untuk menjalankan <i>webserver</i> pada sistem operasi Windows
<i>Web</i>	: kumpulan dokumen multimedia yang berupa <i>text</i> , gambar, suara, animasi, video, yang didalamnya menggunakan protocol Hypertext Transfer Protocol (HTTP).
<i>WebBrowser</i>	: <i>software</i> yang mampu menjalankan dokumen-dokumen <i>web</i> yang memiliki <i>formatHTML</i> , prosesnya yaitu dilakukan oleh komponen <i>webengine</i> yang terdapat pada aplikasibrowser
<i>WebServer</i>	: perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui <i>browser</i> kemudian memberikan tanggapan permintaan dalam bentuk halaman situs <i>web</i> atau lebih umumnya dalam dokumenHTML
<i>While</i>	: digunakan untuk kondisi perulangan dimana

banyaknya perulangan tidak dapat dipastikan pada saat penulisan program

*XAMPP*

: adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program



# INDEKS

## A

Apache, 2, 16, 17, 78, 106, 107, 108  
aplikasi, 1, 8, 16, 17, 73, 82, 84, 89, 95,  
100, 106, 108  
Array, 47, 48, 49, 50, 51, 52, 55, 105  
Asynchronous, 100  
atribut, 3, 5, 8, 22, 68, 69, 89, 96, 97,  
100, 102, 115  
Atribut, 3, 6, 7, 8, 9, 10, 69, 72, 103, 105

## B

bahasa, iii, 1, 2, 7, 15, 18, 21, 23, 31, 39,  
47, 53, 61, 73, 75, 89, 92, 93, 95, 99,  
106, 107  
basis data, 107  
bilangan, 23, 25, 26, 29, 56, 105, 106  
Binary, 74, 105  
body, 2, 3, 5, 6, 7, 9, 11, 18, 19, 20, 26,  
27, 40, 66, 67, 68, 69, 96  
Boolean, 18, 23, 28, 103, 105  
Break, 6, 42, 43, 105  
browser, 1, 2, 3, 4, 5, 8, 9, 16, 54, 70, 71,  
79, 85, 99, 100, 108  
Browser, 1, 16, 108

## C

C++, 89  
canggih, 108  
case sensitive, 3, 13, 18  
class, 90, 91, 92, 94, 97, 98, 102  
*client-side*, 100  
coding, 108

cookie, 84, 85  
Cookies, 84  
CSS, 2, 9, 95, 96, 97, 98, 102, 105

## D

data, 2, 6, 10, 13, 15, 16, 18, 21, 23, 26,  
27, 28, 29, 47, 48, 51, 57, 65, 66, 67,  
68, 70, 72, 74, 79, 80, 81, 83, 84, 85,  
86, 89, 93, 94, 100, 103, 104, 105,  
106, 107, 108  
*database*, 16, 17, 73, 74, 75, 76, 77, 78,  
79, 86, 87, 90, 99, 107, 108  
Define, 105  
dinamis, iii, iv, 6, 15, 16, 65, 106, 107  
*disable*, 84  
dokumen, 1, 2, 3, 4, 5, 8, 9, 16, 20, 95,  
100, 106, 108  
Dokumen, 2, 3, 4, 5  
DOM, 100  
Double, 13, 23, 26, 28, 29, 74, 105  
Do-while, 105

## E

Echo, 64, 105  
editor, 2, 16, 17, 99, 108  
ekstensi, 2, 106  
elemen, 2, 3, 5, 6, 7, 47, 55, 56, 65, 66,  
68, 69, 70, 72, 97, 98, 99, 100, 102,  
105, 115  
email, 74, 82, 90  
encapsulation, 90  
Encapsulation, 89  
error, 36, 82, 86  
event, 100

## F

fungsi, 2, 13, 17, 18, 24, 42, 50, 53, 54,  
55, 58, 59, 60, 61, 62, 70, 73, 75, 76,  
77, 78, 79, 80, 81, 82, 86, 90, 91, 102,  
105, 107

## H

hak akses, 92, 94  
halaman, 2, 4, 5, 6, 7, 8, 9, 15, 16, 17, 57,  
65, 68, 70, 72, 82, 84, 93, 95, 97, 100,  
106, 107, 108  
hardisk, 84  
Head, 4, 12  
*HTML*, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 15,  
16, 17, 18, 20, 56, 65, 95, 96, 97, 98,  
99, 100, 101, 102, 103, 106, 107, 108  
*HTTP*, 1, 16, 105, 106, 107, 108  
*HTTPS*, 16, 106  
Hyperlink, 9, 106

## I

IF, 31, 32, 33, 34, 35, 42, 104, 105, 106  
informasi, 1, 3, 9, 10, 19, 54, 70, 71, 74,  
82, 84, 105, 106, 107  
Inheritance, 89, 92  
Integer, 13, 23, 29, 74, 103, 106  
internet, 1, 73, 106  
Internet, 1, 16, 106  
Internet Explorer, 1, 106

## J

javascript, 99, 101  
*JavaScript*, 2, 99, 100, 106  
jQuery, 100

## K

kode, 2, 7, 15, 16, 17, 19, 34, 35, 39, 40,  
41, 56, 76, 82, 92, 103, 105, 107  
komputer, 1, 15, 16, 57, 59, 73, 78, 90,  
100, 106, 108  
Komputer, 2, 103, 104, 114, 115  
Konqueror, 1

## L

*Layout*, 95  
lisensi, 107  
logika, 25, 27, 31, 32, 33, 34, 35, 42, 106,  
115  
login, 73, 78, 82, 84  
Lynx, 1

## M

maksimum, 23, 106  
markup, 2, 95, 108  
memory, 84  
method, 68, 70, 71, 89, 90, 91, 92, 93  
Mozzila, 1  
MySQL, 5, 6, 16, 73, 74, 75, 76, 77, 78,  
79, 80, 81, 82, 86, 99, 106, 107, 108

## N

Netscape Navigator, 1

## O

objek, 75, 89, 90, 92, 93, 100  
*Objek Oriented Programming*, 92  
*OOP*, 89, 90, 92, 93, 94  
Opera, 1, 16, 106  
Operator, 23, 24, 25, 26, 27, 100

## P

pemograman, 31, 75, 76, 95, 99, 106  
pengkapsulan, 90  
perangkat keras, 2  
perangkat lunak, 2, 16, 106, 107, 108, 109  
percabangan, 31, 33, 34, 35, 36, 38, 106, 108  
platform, 16, 73, 100  
pola, 3, 56, 104  
Polymorphism, 89  
Private, 62, 92, 94  
program, iii, 12, 15, 16, 17, 18, 19, 20, 24, 31, 33, 34, 35, 39, 40, 41, 42, 51, 53, 65, 75, 76, 77, 81, 84, 89, 90, 92, 99, 100, 105, 106, 107, 109  
Programmer, 90, 96  
Property, 90, 91, 93, 94, 97  
Protected, 92, 94  
*Public*, 92, 94

## R

RDBMS, 73, 107  
refresh, 100

## S

Safari, 1, 16  
Script, 17, 20, 22, 29, 36, 37, 38, 42, 51, 52, 62, 86, 93, 97, 98, 99, 100, 107  
scripting, 15, 100, 107  
Selector, 97  
server, 2, 15, 16, 54, 57, 59, 65, 68, 70, 71, 77, 78, 82, 83, 84, 99, 100, 106, 107, 108  
Server, 2, 16, 78, 79, 106, 107, 108  
Services, 106  
sintaks, 15, 23, 31, 39, 47, 53, 58, 85, 89, 95, 99, 100

Sintaks, 77, 78, 97, 100  
software, 1, 2, 108  
String, 20, 23, 25, 26, 27, 28, 55, 61, 64, 74, 75, 100, 103, 108  
struktur, 31, 32, 33, 34, 35, 39, 47, 53, 90, 92, 100, 104, 105, 106  
stuktur, 35, 108  
*switch*, 35, 108  
Sybase, 16, 108  
system, 107, 115

## T

tag, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 17, 22, 56, 71, 95, 96, 99, 103, 105  
Tag, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 99, 100, 103  
Tomcat, 16, 108

## U

URL, 1, 65, 71, 99  
user, 7, 15, 17, 38, 44, 45, 65, 71, 78, 82, 83, 84, 115

## V

value, 4, 47, 48, 49, 54, 55, 68, 69, 70, 80, 87, 97  
variabel, 8, 15, 18, 20, 35, 37, 40, 41, 43, 47, 48, 49, 50, 53, 54, 70, 71, 77, 79, 80, 81, 82, 83, 92, 103, 105, 108  
Variabel, 18, 20, 21, 79, 80, 92, 103

## W

WAMP, 16, 108  
web, iii, iv, 1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 16, 20, 54, 57, 65, 68, 70, 73, 78, 79, 82, 84, 89, 95, 96, 99, 100, 105, 106, 107, 108

While, 39, 40, 41, 43, 62, 108



XAMPP, 16, 17, 78, 103, 109

## BIOGRAFI PENULIS



**Endah Ratna Arumi, S.Kom., M.Cs.** Lulus S1 di Program Studi Teknik *Informatika*, Fakultas Teknik, Universitas Muhammadiyah Magelang tahun 2012, kemudian melanjutkan studi Magisternya (S2) di Program Studi Ilmu Komputer, Universitas Gadjah Mada.

Saat ini adalah Dosen Teknik *Informatika* di Universitas Muhammadiyah Magelang. Penulis aktif mengajar dan mengampu beberapa mata kuliah seperti *Web Programming*, *Artificial Intelligence*, *Intelligence System*, *Algoritm*. Selain mengajar, aktif juga sebagai peneliti dengan fokus penelitian di bidang *Artificial Intelligence*, yang termasuk *Fuzzy logic*, *Decision Support System*, dan *Expert System*. Tidak hanya itu saja, pernah mengikuti berbagai macam konferensi ilmiah baik tingkat nasional ataupun internasional.



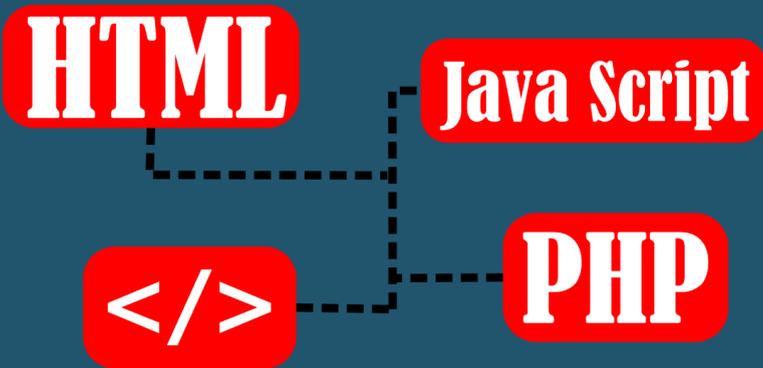
**Setiya Nugroho, ST., M.Eng** Lulus S1 di Program Studi Teknik Elektro, Universitas Diponegoro tahun 2007, kemudian melanjutkan studi Magisternya (S2) di Program Studi Teknologi Informasi, Universitas Gajah Mada.

Saat ini adalah Dosen Teknik *Informatika* di Universitas Muhammadiyah Magelang. Mengampu mata kuliah *Web Programming 2*, *Advance Web Programming*. Selain mengajar, aktif juga sebagai peneliti dengan fokus penelitian di bidang *Integration System*, *Parallel Computing*. Tidak hanya itu saja, pernah mengikuti berbagai macam konferensi ilmiah baik tingkat nasional ataupun internasional seperti *Proceeding 6th University Research Colloquium 2017: Seri Teknologi dan Rekayasa*, *1st Borobudur International Symposium on Applied Science and Engineering (BIS-ASE) 2019 16 October 2019, Magelang, Indonesia*. Motto hidupnya adalah “Berfikir yang baik-baik, Insyallah yang datang yang baik-baik”.



**Pristi Sukmasetya, S.Komp., M.Kom** Lulus S1 di Program Studi Ilmu Komputer IPB University tahun 2015, kemudian melanjutkan studi Magisternya (S2) di Program Studi Magister Ilmu Komputer, Universitas Indonesia. Saat ini menjadi Dosen Teknik Informatika di Universitas Muhammadiyah Magelang.

Aktif mengajar dan mengampu beberapa mata kuliah seperti Human Computer Interaction, Program Building, System Development, Multimedia Concept, Technology Trend, dan Visual Environment Programming. Selain mengajar, aktif juga sebagai peneliti dengan fokus penelitian di bidang human computer interaction, user experience research, *information system*, dan knowledge management. Tidak hanya itu saja, pernah mengikuti berbagai macam konferensi ilmiah baik tingkat nasional ataupun internasional.



# PEMROGRAMAN WEB