



Bab II

Pengenalan PHP

2.1 Apa Itu PHP?

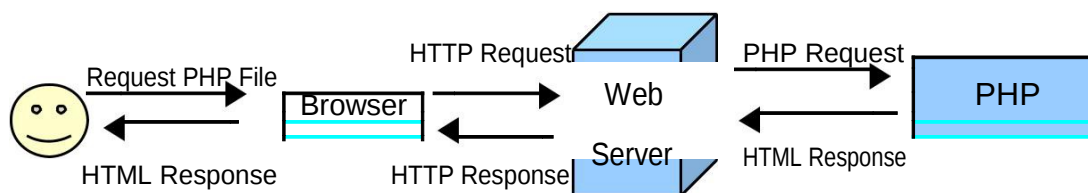
PHP(akronim dari PHP: Hypertext Preprocessor) adalah bahasa serverside scripting yang didesain khusus untuk web. Pada halaman HTML anda dapat menempelkan(embed) kode PHP. Kode PHP dieksekusi di sisi server bukan di komputer klien. Dan hasil yang ditampilkan adalah kode HTML murni.

PHP merupakan hasil kerja seorang bernama Rasmus Lerdorf pada 1995. Namun kemudian PHP berkembang dan tidak hanya merupakan proyek pribadi Rasmus. PHP ditulis ulang dan dengan banyak menambahkan fungsifungsi baru oleh Zeev Suraski dan Andi Gutmants (disingkat Zend) dan lahirlah PHP 3 pada 1998.

PHP masih terus dikembangkan, pada tahun 2002 PHP hadir dengan versi ke 4. PHP 4 dilengkapi dengan Zend Engine dan mengalami banyak peningkatan performa. Dan yang paling akhir pada tahun 2005 PHP 5 hadir dengan dukungan Zend Engine 2. Banyak fitur baru khususnya pada OOP(Object Oriented Programming) dan XML

2.2 Cara Kerja PHP

Cara kerja PHP yang akan kita bahas pada buku ini adalah PHP sebagai bahasa pemrograman untuk mengembangkan aplikasi berbasis web. Karena selain untuk webprogramming PHP juga dapat digunakan untuk mengembangkan aplikasi berbasis desktop dan CLI (Command Line Interface).



Gambar 2.1: Cara kerja PHP

Secara sederhana cara kerja PHP dapat dilihat pada gambar 2.1 diatas. Jika diurut maka proses tersebut terdiri dari:

1. User meminta sebuah halaman PHP
2. Browser mengirim HTTP Request kepada WebServer, misal Apache
3. WebServer mengirim permintaan file PHP tersebut ke PHP processor. PHP processor dapat

- berupa modul(bagian dari webserver) atau terpisah(sebagai CGI/FastCGI)
4. Permintaan diproses oleh PHP processor kemudian hasilnya dikirim kembali ke webserver
 5. Web server memaket kembali hasil tersebut dengan menambahkan HTTP header dan dikirim kembali ke browser.
 6. Browser memproses HTTP packet dan menampilkannya sebagai HTML kepada user.

2.3 Variabel pada PHP

Variabel merupakan elemen yang sangat penting dalam bahasa pemrograman. Hampir setiap bahasa pemrograman mengenal apa itu yang namanya variabel. Variabel sendiri merupakan suatu bentuk penyimpanan data sementara pada memori komputer yang akan diolah lebih lanjut. Contoh variabel pada PHP antaran lain:

```
$nama = 'Dayat Miftahul Jannah';
$angka_1 = 1;
$angka_2 = 2;
$hasil = $angka_1 + $angka_2;
```

Ketentuanketentuan dalam membuat variabel:

- Variabel dapat terdiri dari huruf, angka dan underscore(_) dan tentu saja tanda dollar \$.
- Variabel tidak dapat diawali dengan angka.
- Variabel bersifat case sensitive artinya membedakan huruf kecil dan huruf besar.
- Variabel \$nama tidak sama dengan \$NaMa. Untuk itu hatihati dalam penulisan nama variabel.

Pada variabel anda dapat mengubah nilainya, contoh \$angka_1 = 1 dapat anda ubah menjadi \$angka_1 = 2 saat suatu script dijalankan. Ini berbeda dengan konstanta(akan dibahas berikutnya).

2.3.1 Tipe Variabel

Pada PHP kita tidak perlu mendeklarasikan tipe variabel secara eksplisit, istilah kerennya *dynamic typing*. Karena PHP secara otomatis dapat menentukan tipe variabel berdasarkan nilai yang ada pada variabel tersebut. Berikut ini adalah beberapa tipe data yang didukung oleh PHP.

Tabel 2.1: Macammacam Tipe Data

TipeData	Keterangan
Integer	Digunakan untuk semua angka
String	Digunakan untuk semua huruf, angka, spasi, dan simbol
Double	Digunakan untuk bilangan real
Boolean	Digunakan untuk nilai True atau False
Array	Digunakan untuk menampung beberapa data sekaligus
Object	Digunakan untuk class

2.3.2 Konstanta

Hampir sama dengan variabel, konstanta juga digunakan untuk penyimpanan nilai sementara. Namun perbedaan konstanta dengan variabel adalah pada konstanta anda tidak dapat mengubah

nilainya jika sudah dideklarasikan. Cara pendeklarasiannya pun berbeda dengan variabel. Pada konstanta digunakan keyword `define` untuk mendeklarasikan variabel. Konstanta juga tidak diawali dengan tanda \$ (dollar).

```
define('HARGA', 1500);  
define('NAMA', 'Dayat Miftahul Jannah');
```

Dapat dilihat pada kode diatas bahwa kita selalu gunakan huruf KAPITAL untuk konstanta. Hal ini tidak harus dilakukan namun semacam peraturan tidak tertulis jika konstanta maka sebaiknya gunakan huruf kapital.

2.3.3 Penerapan Teori Variabel dan Konstanta

1. Jika belum silahkan buat folder *bab2* didalam folder *htdocs/webpro*.
2. Pastikan Apache sudah berjalan.
3. Buka gedit (Linux)/Notepad++ (Windows)
4. Ketik kode berikut:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1transitional.dtd">  
3. <html xmlns="http://www.w3.org/TR/xhtml1">  
4. <head>  
5.   <title>Variabel dan Konstanta</title>  
6. </head>  
7. <body>  
8. <?php  
9.   $menu = 'Menu: Nasi Pecel';  
10.  define('HARGA', 2500);  
11.  
12.  $menu = 'Menu: Nasi Rawon';  
13.  define('HARGA', 5000);  
14.  
15.  echo $menu . '<br />';  
16.  echo HARGA;  
17. ?>  
18. </body>  
19. </html>
```

5. Simpan dengan nama *variabel.php* pada folder *bab2*
6. Arahkan browser anda ke alamat <http://localhost/belajarweb/bab2/variabel.html>
7. Output dari file tersebut adalah

Menu: Nasi Rawon
2500

Klik kanan browser anda lalu klik View Source (untuk Firefox) untuk melihat kode HTML dari halaman tersebut. Perhatikan bahwa tag `<?php` dan `?>` yang ada hanyalah kode HTML murni.

Penjelasan Script variabel.php

Jika anda perhatikan, sama seperti pada bab sebelumnya kita hanya menulis sebuah file html biasa.

Namun ada sedikit perbedaan, dimana pada file variabel.php terdapat sebuah tag baru yaitu `<?php` dan `?>`. Script dalam tag inilah yang akan diproses oleh PHP processor. Semua tulisan diluar tag `<?php` dan `?>` akan diabaikan oleh PHP.

Cara seperti yang kita lakukan pada file variabel.php adalah menempelkan PHP pada HTML. Jika kumpulan mp3 anda adalah western song mungkin anda lebih suka mendengarnya dengan sebutan embedding PHP inside HTML :).

Jika anda masih ingat teori sebelumnya tentang variabel, dapat dilihat bahwa variabel `$menu` kita deklarasi ulang dan nilainya diubah menjadi 'Menu: Nasi Rawon' (Baris: 12). Namun hal tersebut tidak berlaku pada konstanta, pada baris 13 kita deklarasi ulang konstanta `HARGA`. Nilai dari konstanta tersebut tidak berubah.

Hal itu ditunjukkan ketika kita mencetak nilai dari `$menu` dan `HARGA`. `$menu` berubah menjadi 'Nasi Rawon' dan konstanta tetap 2500.

Pada PHP untuk mencetak sesuatu dapat digunakan perintah `echo`. Contoh penggunaan dapat dilihat pada baris 15 dan 16. Pada konstanta tidak perlu diberi tanda `$`.



TIPS:

Selain perintah `echo` anda juga dapat menggunakan perintah `print` untuk mencetak output.

2.4 Komentar

Komentar pada script bertujuan untuk memberitahu pembacanya, baik orang lain ataupun anda sendiri. Biasanya komentar digunakan untuk menjelaskan tujuan dituliskannya script tersebut, siapa penulisnya, kapan ditulis dan sebagainya. Komentar juga berguna bagi anda sendiri ketika suatu waktu anda lupa mengapa anda menulis file ini, fungsifungsinya untuk apa dan banyak lainnya.

PHP akan mengabaikan semua text didalam komentar. Jadi tidak akan berpengaruh pada jalannya suatu script.

PHP mengenal tiga jenis gaya komentar. Pertama model bahasa C (banyak baris/multi line), biasanya dapat diletakkan pada bagian atas script. Lihatlah script dibawah ini.

```
1. <?php
2.
3. /*
4.  Filename : db.php
5.  Author   : Dayat Miftahul Jannah
6.  Created  : 20022018 18:22
7.  Updated  : 31102018 16:11
8.  Desc     : File yang menyimpan konfigurasi untuk koneksi ke MySQL Database
9. */
10.
11. ?>
```

Yang kedua adalah model C++, seperti yang ditunjukkan dibawah ini.

```
echo $nama;    // cetak nama
```

Yang ketiga adalah model shell script.

```
echo HARGA;      # cetak harga
```

Sekarang coba modifikasi file variabel.php tersebut dengan menambahkan komentar lalu lihat hasilnya. Apakah masih sama atau tidak.

2.5 Operator

Operator merupakan simbol yang dapat digunakan untuk memanipulasi nilai dan variabel. Pada bagian sebelumnya kita sudah menggunakan beberapa operator diantaranya =, ==, <, >, +, * dan lainnya. Selanjutnya kita akan membahas operator-operator ini dan operator lainnya lebih detail lagi.

2.5.1 Operator Aritmatik

Pasti anda sudah banyak mengenal berbagai macam operator aritmatik. Jika anda pernah mengenyam pendidikan yang bernama Sekolah Dasar atau SD pasti anda sering menulis operator-operator ini pada buku anda :).

Tabel 2.2: Operator Aritmatik

Operator	Nama	Contoh
+	Penambahan	\$a + \$b
	Pengurangan	\$a - \$b
*	Perkalian	\$a * \$b
/	Pembagian	\$a / \$b
%	Modulus	\$a % \$b

Yang mungkin bapak atau Ibu guru SD kita lupa mengajarkan adalah tentang modulus. Modulus merupakan sisa pembagian dari dua bilangan. Lihat contoh berikut untuk lebih jelas.

```
1. <?php
2.
3. $bil1 = 10;
4. $bil2 = 5;
5. $bil3 = 3;
6.
7. echo $bil1 % $bil2; // output 0
8. echo $bil1 % $bil3; // output 1
9. echo $bil2 % $bil3; // output 2
10.
11. ?>
```

2.5.2 Operator Kombinasi

Saya sendiri bingung menjelaskan operator ini, saya takut katakata saya menyesatkan pikiran anda :). Jadi lebih baik kita langsung lihat contoh saja biar lebih jelas.

Tabel 2.3: Operator Kombinasi

Operator	Penggunaan	Sama dengan
+=	\$a += \$b	\$a = \$a + \$b
=	\$a = \$b	\$a = \$a \$b

<code>*=</code>	<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>
<code>/=</code>	<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>
<code>%=</code>	<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>

Penggunaan operator kombinasi memang favorit para programmer karena memang mereka ini orang-orang yang cenderung malas. Waduh, saya juga termasuk kelihatannya :).

2.5.3 Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua nilai. Penggunaan ekspresi ini juga akan menghasilkan nilai true atau false tergantung dari perbandingan.

Tabel 2.4: Operator perbandingan

Operator	Nama	Penggunaan
<code>==</code>	sama dengan	<code>\$a == \$b</code>
<code>===</code>	identik (harus bertipe sama)	<code>\$a === \$b</code>
<code>!=</code>	tidak sama dengan	<code>\$a != \$b</code>
<code><></code>	tidak sama dengan	<code>\$a <> \$b</code>
<code><</code>	kurang dari	<code>\$a < \$b</code>
<code>></code>	lebih dari	<code>\$a > \$b</code>
<code><=</code>	kurang dari atau sama dengan	<code>\$a <= \$b</code>
<code>>=</code>	lebih dari atau sama dengan	<code>\$a >= \$b</code>

Operator perbandingan biasanya digunakan pada struktur kontrol seperti percabangan dan perulangan.

2.5.4 Operator Logika

Operator logika digunakan untuk melakukan tes terhadap sebuah kondisi logika dari dua atau lebih perbandingan. Sebagai contoh, kita mungkin menemukan kasus nilai dari variabel `$a` berada diantara 0 sampai 10. Untuk mengetes kondisi `$a >= 0` dan `$a <= 10`, kita gunakan operator AND.

Tabel 2.5: Operator Logika

Operator	Nama	Penggunaan	Keterangan
<code>!</code>	NOT	<code>!\$a</code>	TRUE jika <code>\$a</code> adalah FALSE, berlaku sebaliknya
<code> </code>	OR	<code>\$a \$b</code>	TRUE jika <code>\$a</code> atau <code>\$b</code> bernilai TRUE atau keduanya bernilai TRUE
<code>&&</code>	AND	<code>\$a && \$b</code>	TRUE jika <code>\$a</code> dan <code>\$b</code> bernilai TRUE
<code>xor</code>	XOR	<code>\$a xor \$b</code>	TRUE jika <code>\$a</code> atau <code>\$b</code> bernilai TRUE, tapi tidak keduanya

2.5.5 Operator Increment/Decrement

Operator increment/decrement digunakan untuk menambah/mengurangi nilai dari suatu variabel dengan satu. Increment memiliki simbol `++` (double plus) sedangkan decrement (double minus).

Tabel 2.5: Operator Increment/Decrement

Nama	Contoh	Keterangan
PreIncrement	++\$a	Tambah \$a dengan satu, lalu kembalikan nilainya
PostIncrement	\$a++	Kembalikan nilai \$a, lalu tambah \$a dengan satu
PreDecrement	\$a--	Kurangi \$a dengan satu, lalu kembalikan nilainya
PostDecrement	\$a--	Kembalikan nilai \$a, lalu kurangi \$a dengan satu

Mungkin akan lebih jelas jika anda melihat contoh dibawah ini.

```

1. <?php
2.
3. echo "<h3>Postincrement</h3>";
4. $a = 5;
5. echo "Seharusnya 5: " . $a++ . "<br />";
6. echo "Seharusnya 6: " . $a . "<br />";
7.
8. echo "<h3>Preincrement</h3>";
9. $a = 5;
10. echo "Seharusnya 6: " . ++$a . "<br />";
11. echo "Seharusnya 6: " . $a . "<br />";
12.
13. echo "<h3>Postdecrement</h3>";
14. $a = 5;
15. echo "Seharusnya 5: " . $a . "<br />";
16. echo "Seharusnya 4: " . $a . "<br />";
17.
18. echo "<h3>Predecrement</h3>";
19. $a = 5;
20. echo "Seharusnya 4: " . $a . "<br />";
21. echo "Seharusnya 4: " . $a . "<br />";
22.
23. ?>

```

2.5.6 Operator String

Sebenarnya pada contohcontoh sebelumnya operator string ini sudah sering kita gunakan. Operator string yang dimaksud adalah tanda "." (titik). Tanda titik ini dapat digunakan untuk menggabungkan string.

```

1. <?php
2.
3. $a = 'LUG';
4. $b = 'STIKOMP';
5. $c = $a . ' ' . $b; // hasil: LUG STIKOMP
6.
7. // atau dengan kombinasi
8. $a .= ' STIKOMP'; // hasil: LUG STIKOMP
9.
10. ?>

```

2.6 Escape Character

Pada saat menulis kode anda akan sangat sering menjumpai suatu kondisi dimana kita harus mencetak tanda petik baik petik satu maupun petik dua. Permasalahannya tanda tersebut sudah

digunakan sebagai penanda untuk mencetak string. Lalu, bagaimana pemecahannya? Ada beberapa solusi untuk mencetak petik didalam petik itu sendiri:

1. Gunakan petik satu (') sebagai penutup string jika ingin mencetak petik dua (") dan sebaliknya.
2. Menggunakan Escape Character \ (backslash)

Berikut ini adalah contoh penyelesaian dari kedua solusi diatas.

```
<?php

/*
SOLUSI No. 1
=====
    Jika ingin mencetak petik satu gunakan petik dua sebagai penutup string
    Jika ingin mencetak petik dua gunakan petik satu sebagai penutup string
*/
$a = 'John Says: "My dear, I Love You."<br />';
$b = "The Girl Says: 'Go to Hell...!!!'<br />";
echo $a . $b . '<br />';

/*
SOLUSI No. 2
=====
    Gunakan escape character \ (backslash)
*/
$a = "John Says: \"My dear, I Love You.\"<br />";
$b = 'The Girl Says: \'Go to Hell...!!!\'<br />';
echo $a . $b;

?>
```

2.7 Special Character

Pembahasan ini sebenarnya tidak begitu krusial jika dihubungkan dengan pembuatan tampilan web. Namun akan sangat berguna dalam proses pencarian kesalahan ketika kita mendesain suatu halaman website. Bingung? Sama saya juga bingung dengan apa yang saya tulis :p.

Pada intinya HTML mengabaikan semua karakter whitespace (spasi, tab, baris baru) dan menggantinya hanya dengan satu spasi ketika ditampilkan. Special character disini mencakup

- Baris baru (\n)
- Tab (\t)

Sebenarnya masih banyak karakter spesial lainnya seperti \r, \0, \b dan lainlain. Untuk lebih jelasnya cobalah contoh berikut.

```
1. <?php
2.
3. echo "INI\t\tBANYAK\t\tSPASI<br />";
4. echo "INI BARIS SATU<br />";
5. echo "INI BARIS DUA<br />";
6. echo "INI BARIS TIGA<br />";
7.
8. echo "\n\n<br />\n\n";
9. echo "INI BARIS SATU<br />\nINI BARIS DUA<br />\nINI BARIS TIGA";
```

10.

11. ?>

Output dari kode diatas kurang lebih seperti berikut:

```
INI BANYAK SPASI  
INI BARIS SATU  
INI BARIS DUA  
INI BARIS TIGA  
  
INI BARIS SATU  
INI BARIS DUA  
INI BARIS TIGA
```

Jika dilihat dari browser tidak ada yang berbeda dari group 1 dan group 2 yang ada pada kode diatas. Tapi jika anda melihat source HTML dengan cara klik kanan → View Source (Firefox) akan terlihat perbedaannya.



TIPS:

Untuk mencetak special character seperti \n, \t, \r dan sebagainya harus menggunakan petik dua sebagai penutup string.

Ringkasan Bab II

Pada Bab II kita telah membahas sejarah singkat tentang PHP yang diciptakan oleh Rasmus Lerdorf dan kemudian dikembangkan oleh Zend. Kita juga telah membahas hal-hal dasar seputar PHP yang meliputi:

- Variabel
- Konstanta
- Operator
- Special Character