

## **BAB VII**

### **DASAR-DASAR PHP**

#### **A. KOMPETENSI DASAR**

- Memahami struktur dasar dokumen PHP.
- Mampu membuat dokumen PHP yang baik dan benar.
- Mampu memanfaatkan elemen-elemen dasar untuk mengolah dan menampilkan informasi.

#### **B. ALOKASI WAKTU**

4 JS (4 x 50 menit)

#### **C. PETUNJUK**

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

#### **D. DASAR TEORI**

##### **1. PHP**

PHP (*PHP: Hypertext Preprocessor*) merupakan salah satu bahasa *web-scripting* yang sangat *powerful*. Sejak pertama kali diperkenalkan, bahasa ini dimaksudkan untuk menghasilkan halaman-halaman web yang dinamis. Hingga saat ini, PHP banyak sekali digunakan dalam membuat aplikasi web—baik lokal maupun Internet—dinamis dan atraktif.

##### **2. Tipe Data**

PHP mendukung delapan tipe primitif, yang terdiri dari empat tipe skalar (*boolean*, *integer*, *float/double*, dan *string*), dua tipe gabungan (*array* dan *object*), dan sisanya adalah tipe khusus (*NULL* dan *resource*). Meskipun keberadaan tipe data dalam deklarasi variabel tidak penting, namun dalam tahap selanjutnya sangatlah penting.

##### **3. Komentar**

PHP mendukung tiga jenis sintaks untuk menuliskan baris komentar, meliputi *style C*, *C++*, dan *Perl/Shell*.

```
/* awal
Komentar style C
akhir */

// Komentar style C++

# Komentar style Perl/Shell
```

Meskipun ketiga jenis style di atas valid, namun upayakan untuk memilih style yang paling sesuai. Umumnya, style C digunakan untuk komentar yang terdiri dari beberapa baris, sedangkan sisanya untuk satu baris.

#### 4. Tag PHP

Parser PHP bekerja dengan menguraikan instruksi-instruksi yang diletakkan di antara tag pembuka dan penutup. Ada empat jenis style tag yang dapat kita gunakan untuk menyatakan bahwa kode merupakan instruksi PHP.

Dua dari empat jenis tag yang selalu tersedia tanpa memerlukan konfigurasi adalah style XML dan SCRIPT.

```
<?php
    echo 'style XML';
?>

<script language="php">
    echo 'style SCRIPT';
</script>
```

Dua style tag lainnya memerlukan aktivasi dukungan di file konfigurasi php.ini.

```
<?
    // Aktifkan dukungan style SGML/Short di php.ini
    // short_open_tag = On

    echo 'style SGML';
?>

<%
    // Aktifkan dukungan style ASP di php.ini
    // asp_tags = On

    echo 'style ASP';
%>
```

## E. LATIHAN

### 1. Program PHP

Dalam pembuatan aplikasi web, instruksi-instruksi PHP akan dikombinasikan dengan elemen-elemen HTML. Meskipun PHP dapat digunakan untuk membungkus keseluruhan dokumen HTML, namun sebaiknya hal ini tidak kita lakukan. Selain dapat menyulitkan pembacaan kode, langkah ini juga kurang efisien.

Pendekatan yang disarankan dalam kombinasi ini adalah memisahkan kode program PHP dari dokumen HTML. Sebagai contoh, ketikkan kode program berikut dan simpan dengan nama **php\_html.php** (ingat, ekstensi yang digunakan adalah **php**).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Demo</title>
</head>

<body>

<?php
echo 'Kode PHP di sini';
// ...
?>

<p> Dokumen HTML </p>

<?php
echo 'Kode PHP di sini';
// ...
?>

</body>
</html>
```

Berkeanaan dengan kombinasi ini, tag-tag PHP juga menyediakan shortcut untuk meringkas penulisan kode. Shortcut ini umumnya digunakan ketika ingin menuliskan kode-kode kecil di dalam HTML.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Demo</title>
</head>

<body>

<p> Kode <?php echo "PHP";?> di HTML </p>

</body>
</html>
```

## 2. Variabel

Sebagaimana diketahui, variabel diidentifikasi melalui karakter dollar (\$) dan diikuti nama variabel. PHP mengizinkan kita untuk mendeklarasikan dan meng-*assign* suatu nilai sekaligus pada saat variabel akan digunakan.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Demo Variabel</title>
</head>

<body>

<?php

// Deklarasi dan inisialisasi
$bil = 3;

echo $bil;

?>

</body>
</html>
```

Di tahap pengembangan, kita bisa memanfaatkan fungsi `var_dump()` atau `print_r()` untuk memudahkan pemeriksaan variabel.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Demo Variabel</title>
</head>

<body>

<?php

// Deklarasi dan inisialisasi
$bil = 3;

// Dumping informasi mengenai variabel
var_dump($bil);

print_r($bil);

?>

</body>
</html>
```

Amati dan pahami hasil keluaran dari fungsi `var_dump()`. Apa bedanya `var_dump()` dengan konstruksi bahasa `echo`? Jelaskan!



Perlu sekali diperhatikan, nama variabel di PHP bersifat case-sensitive. Dengan demikian, variabel `$var` berbeda dengan `$var` atau `$VAR`.

Apabila Anda ingin memeriksa apakah suatu variabel di-set atau tidak, gunakan konstruksi bahasa `isset()`. Konstruksi bahasa ini juga dapat digunakan untuk memeriksa beberapa variabel sekaligus.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Demo Variabel</title>
</head>
<body>

<?php

$bil = 3;
var_dump($bil);
// Output: int(3)

$var = "";
var_dump($var);
// Output: string(0) ""

$var = null;
var_dump($var);
// Output: NULL

?>

</body>
</html>
```

### 3. Tipe Data dan Casting

PHP menyediakan fungsi-fungsi berawalan `is_` yang dapat dimanfaatkan untuk menguji tipe data suatu variabel.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Cek Tipe</title>
</head>
<body>

<?php

$bil = 3;
var_dump(is_int($bil));
// Output: bool(true)

$var = "";
var_dump(is_string($var));
// Output: bool(true)

?>

</body>
</html>
```

Kecuali fungsi `is_bool()`, penamaan fungsi-fungsi lainnya persis seperti tipe data terkait, misalnya `is_integer()`, `is_float()`, `is_object()`, dan `is_null()`.

## Casting

Untuk menguji aplikasi web, kita mempublikasikannya ke web server, baik secara lokal maupun Internet. Lingkungan lokal tentu merupakan pilihan yang efisien, khususnya ketika aplikasi masih dalam tahap pengembangan.

Operator	Deskripsi
(int), (integer)	Casting ke integer
(double), (float), (real)	Casting ke floating point
(string)	Casting ke string
(array)	Casting ke array
(object)	Casting ke object
(bool), (boolean)	Casting ke boolean
(unset)	Casting ke null, seperti pemanggilan unset

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Casting Tipe</title>
</head>
<body>

<?php

$str = '123abc';

// Casting nilai variabel $str ke integer
$bil = (int) $str; // $bil = 123

echo gettype($str);
// Output: string

echo gettype($bil);
// Output: integer

?>

</body>
</html>
```

## 4. Pernyataan Seleksi

Pernyataan seleksi untuk pengambilan keputusan merupakan bagian dari bahasa pemrograman yang sangat penting. Pernyataan-pernyataan pengambilan keputusan di PHP diklasifikasikan ke dalam empat bagian: `if`, `if-else`, `if-elseif`, dan `switch`.

- Pernyataan `if`

Pernyataan `if` terdiri dari suatu ekspresi dan sebuah statemen atau blok statemen yang dieksekusi apabila ekspresi bernilai `true`.

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <title>Seleksi</title>
</head>
<body>

<?php

$a = 10;
$b = 5;

if ($a > $b) {
    echo 'a lebih besar dari b';
}

?>

</body>
</html>
```

- **Pernyataan if-else**

Pernyataan ini sebenarnya merupakan ekspansi dari pernyataan `if`. Di sini blok pertama akan dieksekusi manakala ekspresi bernilai `true`, sebaliknya—jika ekspresi bernilai `false`—blok kedua yang dieksekusi.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Seleksi if-else</title>
</head>
<body>

<?php

$a = 10;
$b = 5;

if ($a > $b) {
    echo 'a lebih besar dari b';
} else {
    echo 'a TIDAK lebih besar dari b';
}

?>

</body>
</html>
```

- **Pernyataan if-elseif**

Pernyataan ini sebenarnya merupakan ekspansi dari `if-else`, di mana di ditambahkan lagi blok `if-elseif`. Bentuk pernyataan `if-elseif` memungkinkan kita untuk menciptakan seleksi yang lebih kompleks.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Seleksi if-elseif</title>
</head>
<body>
```

```
<?php
$a = 10;
$b = 5;

if ($a > $b) {
    echo 'a lebih besar dari b';
} elseif ($a == $b) {
    echo 'a sama dengan b';
} else {
    echo 'a kurang dari b';
}

?>

</body>
</html>
```

- **Pernyataan switch**

Pernyataan **switch** merupakan sebuah pernyataan *control flow* yang dimulai dengan suatu ekspresi dan mentransfer kontrol ke satu kasus berdasarkan nilai ekspresi.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Seleksi switch</title>
</head>
<body>

<?php
$i = 0;

if ($i == 0) {
    echo "i equals 0";
} elseif ($i == 1) {
    echo "i equals 1";
} elseif ($i == 2) {
    echo "i equals 2";
}

// Ekuivalen, dengan pendekatan switch
switch ($i) {
case 0:
    echo "i equals 0";
    break;
case 1:
    echo "i equals 1";
    break;
case 2:
    echo "i equals 2";
    break;
}

?>

</body>
</html>
```



## 5. Pengulangan

Bagian ini akan menguraikan jenis-jenis dan implementasi struktur kontrol pengulangan.

- Pengulangan while

Pada pernyataan ini, ekspresi akan dievaluasi dan pengulangan dieksekusi jika dan hanya jika ekspresi bernilai true.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Loop while</title>
</head>
<body>

<?php
$i = 0;

while ($i < 10) {
  echo $i;

  // Inkremen counter
  $i++;
}

?>

</body>
</html>
```

- Pengulangan do-while

Bentuk pengulangan ini mirip dengan while, kecuali bahwa ekspresi pengontrolan pengulangan dilakukan di akhir blok. Ini juga berarti bahwa blok pengulangan akan dieksekusi sedikitnya satu kali, meskipun ekspresi bernilai false.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Loop do-while</title>
</head>
<body>

<?php
$i = 0;

do {
  echo $i;

  // Inkremen counter
  $i++;
} while ($i < 10);

?>

</body>
</html>
```

- Pengulangan for

Pernyataan pengulangan ini paling banyak digunakan di dalam program, khususnya ketika jumlah iterasinya sudah diketahui.

Struktur pengulangan `for` terdiri dari tiga bagian: ekspresi inisialisasi, kondisi perulangan, dan ekspresi inkremen.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Loop for</title>
</head>
<body>

<?php

for ($i = 0; $i < 10; $i++) {
  echo $i;
}

?>

</body>
</html>
```

- Pengulangan foreach

Sebagai tambahan, PHP juga menyediakan konstruksi `foreach` yang dapat digunakan untuk melakukan iterasi di array atau koleksi.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Loop foreach</title>
</head>
<body>

<?php

$arr = array(1, 2, 3, 4);

foreach ($arr as $value) {
  echo $value;
}

?>

</body>
</html>
```

## 6. Fungsi dan Prosedur

Keberadaan fungsi/prosedur sangat membantu dalam mengorganisir kode program dan menerapkan aspek guna ulang. Di PHP, pembuatan fungsi dan prosedur menggunakan sintaks sama, di mana yang membedakan hanya pengembalian nilai.

- Definisi Fungsi/Prosedur

Contoh definisi fungsi dan prosedur beserta cara pemanggilannya diperlihatkan sebagai berikut:

```
<?php
// Contoh prosedur
function do_print() {
    // Mencetak informasi timestamp
    echo time();
}

// Memanggil prosedur
do_print();

echo '<br />';

// Contoh fungsi penjumlahan
function jumlah($a, $b) {
    return ($a + $b);
}

echo jumlah(2, 3);
// Output: 5
?>
```

- Argumen Fungsi/Prosedur

Suatu fungsi dapat memiliki nol atau lebih argumen. Adapun jika diperlukan, juga bisa dideklarasikan argumen yang sifatnya opsional. Deklarasi ini sekaligus menginisialisasi nilai default pada argumen. Selain itu, argumen opsional harus diletakkan di urutan paling belakang.

```
<?php

/**
 * Mencetak string
 * $teks nilai string
 * $bold adalah argumen opsional
 */
function print_teks($teks, $bold = true) {
    echo $bold ? '<b>' . $teks . '</b>' : $teks;
}

print_teks('Indonesiaku');
// Mencetak dengan huruf tebal

print_teks('Indonesiaku', false);
// Mencetak dengan huruf reguler
?>
```

## **F. STUDI KASUS**

1. Buat fungsi greeting yang menerima argumen integer jam (format 24 jam) dan mengembalikan nilai string (Selamat Pagi, Selamat Siang, atau Selamat Malam). Jika nilai tidak memenuhi maka akan mengembalikan string kosong.

### G. TUGAS PRAKTIKUM

1. Buat program sederhana untuk men-*generate* sel tabel secara fleksibel. Tekniknya, buat sebuah fungsi yang menerima argumen berupa jumlah sel dan jumlah kolom. Jadi, pembentukan sel tabel didasarkan pada nilai jumlah sel dan jumlah kolom yang diberikan. Sebagai contoh, hasil program dengan argumen 12 (sel) dan 3 (kolom) diperlihatkan seperti Gambar 1.

1	2	3
4	5	6
7	8	9
10	11	12

Gambar 1. Pembuatan sel dengan argumen 12 dan 3

Untuk contoh argumen 10 dan 4 diperlihatkan pada Gambar 2.

1	2	3	4
5	6	7	8
9	10		

Gambar 2. Pembuatan sel dengan argumen 10 dan 4