

Modul 4 : Dasar PHP

4.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- 50 menit **pengayaan**

4.2 Tujuan

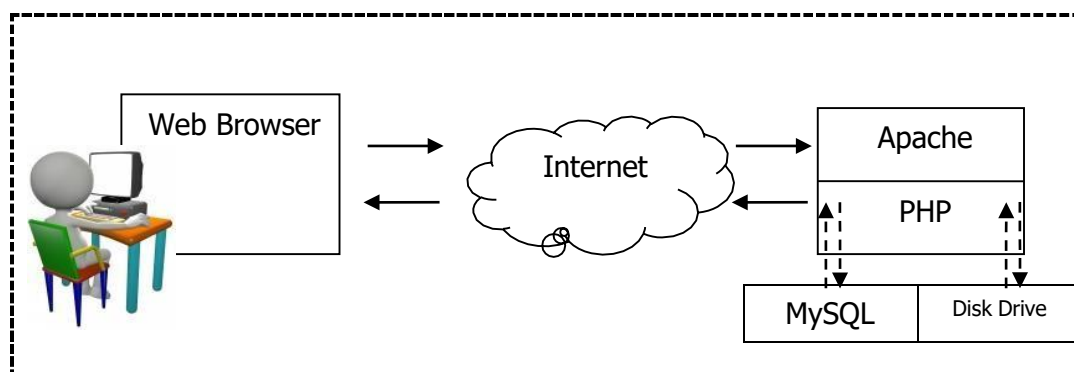
Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami gambaran umum aplikasi web.
2. Memahami kerangka program PHP.
3. Memahami tipe data, variabel, dan operator pada PHP.
4. Memahami sintaks untuk mengatur alur program PHP.

4.3 Dasar Teori

a) PHP

PHP adalah salah satu bahasa pemrograman yang dirancang untuk membangun aplikasi **web dinamis**. PHP juga disebut sebagai **server-side programming/scripting** yang berarti *script/program* yang berjalan di sisi *server*. Kode PHP di-*parse* di *web server* oleh **interpreter** dan diterjemahkan ke dalam bentuk HTML (*Hypertext Markup Language*) dan pada akhirnya dikirimkan dan ditampilkan pada *browser* yang dibuka oleh *user*. Gambar 4.1 memberikan gambaran singkat alur kerja PHP dalam *web server*.



Gambar 4.1 Alur request-response PHP

Sintaks kode PHP biasanya diletakkan di dalam HTML, walaupun bisa juga dibuat dokumen dengan PHP murni. Kode PHP diawali dengan **tag pembuka PHP** “<?php” dan diakhiri dengan **tag penutup**

PHP “>” atau bisa juga dengan bentuk yang lebih singkat yaitu dengan awalan “<?” dan diakhiri “>”. Kode PHP harus disimpan di dalam file berekstensi .php. Jika tidak, maka kode PHP tidak akan dieksekusi oleh *web server* walaupun ditulis di antara *tag* pembuka dan penutup PHP. Berikut satu contoh kode PHP sederhana.

```
1 <?php
2     echo "Hello World";
3     echo 10000;
```

Perintah `echo` berfungsi untuk mencetak nilai baik teks maupun numerik ke *browser*. *Keyword-keyword* perintah PHP seperti `echo` dan juga tag pembuka dan penutup PHP (`<?php` dan `>`) tidak akan ikut tercetak di *browser*. Selain `echo`, kita juga bisa menggunakan perintah `print`:

```
1 <?php
2     print "Hello World";
```

Jika suatu file PHP hanya berisi kode PHP saja tanpa HTML atau kode PHP terletak di akhir *file*, maka *tag* penutup PHP (`>`) tidak perlu ditulis. Contoh:

```
1 <?php
2     print "Hello World";
3     <!-- kode HTML -->
4 </html>
5
6 <?php
7     print "Hello World";
8     // akhir file
```

Setiap perintah atau statemen di dalam kode PHP harus diakhiri dengan tanda titik koma (“;”).

Ada beberapa cara untuk meletakkan kode PHP dalam sebuah dokumen, yaitu dengan menyisipkan PHP dalam HTML atau HTML dalam PHP. Berikut contoh PHP dalam HTML.

```
1 <html>
2 <head>
3     <title>PHP dalam HTML</title>
4 </head>
5 <body>
6     <?php echo "<p>Teks dari kode PHP</p>"; ?>
7 </body>
8 </html>
```

Berikut contoh HTML dalam PHP.

```
1 <?php
2     echo "<html>";
3     echo "<head>";
4     echo "<title>HTML dalam PHP</title>";
5     echo "</head>";
6     echo "<body>";
7     echo "<p>Paragraf 2: Teks dari kode PHP</p>";
8     echo "</body>";
```

b) Tipe Data, Variabel, dan Operator

PHP mendukung delapan tipe data. Empat nilai skalar (*single-value*): *integer*, *float*, *string*, dan *boolean*; nilai koleksi: *array*. Sisanya adalah tipe khusus yaitu *resource* dan *null*.

| | |
|----------------|---|
| Integer | <i>Integer</i> adalah tipe data yang merepresentasikan bilangan utuh atau bulat, positif maupun negatif dan bukan pecahan. Rentang dari bilangan integer adalah -2.147.483.648 sampai 2.147.483.647. |
| Float | <i>Float</i> disebut sebagai bilangan pecahan atau sering juga disebut sebagai bilangan <i>real</i> yang mengandung angka desimal di belakang koma. Contohnya adalah 3,14; 2,54; dan 9,99. |
| String | PHP mendukung tipe data <i>string</i> yaitu tipe data karakter yang biasanya diekspresikan dengan diapit oleh petik ganda atau petik tunggal. Contoh: <code>'ini adalah string' atau "ini adalah string"</code> |
| Boolean | Tipe <i>boolean</i> adalah tipe data yang merepresentasikan nilai kebenaran (<i>truth value</i>). Nilai yang mungkin pada tipe <i>boolean</i> adalah <i>true</i> dan <i>false</i> . Penulisan nilai <i>true</i> dan <i>false</i> bersifat <i>case-insensitive</i> sehingga <code>"true"</code> , <code>"True"</code> , dan <code>"TRUE"</code> adalah sama dan <code>"false"</code> , <code>"False"</code> , dan <code>"FALSE"</code> adalah sama. Contoh: <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre><?php \$found = true; if (\$found) { echo "Data telah ditemukan"; }</pre> </div> <p>Maksud dari kode diatas adalah jika variabel <code>\$found</code> bernilai <i>true</i>, maka dilakukan kode di dalam blok <code>if</code> akan dieksekusi.</p> |
| Array | <i>Array</i> adalah variabel yang menyimpan sekelompok nilai yang dapat diidentifikasi berdasarkan posisinya atau indeksnya. Contoh: <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre>\$person = array('Dadang', 'Jajang', 'Udin', 'Rendra');</pre> </div> <p>Penulisan <i>array</i> juga dapat dilakukan dengan sintaks yang lebih ringkas. Contoh:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre>\$person = ['Dadang', 'Jajang', 'Udin', 'Rendra'];</pre> </div> |

Pendefinisian Variabel

Variabel di PHP ditulis dengan awalan karakter dolar (\$). Contoh:

| | |
|---|-------------------------------------|
| 1 | <code>\$username = "Hendro";</code> |
| 2 | <code>\$usia = 25;</code> |

PHP tidak seperti bahasa pemrograman lain yang mengharuskan kita untuk memberikan atau menuliskan tipe variabel. Artinya PHP mempunyai kemampuan untuk mendeteksi secara otomatis apa tipe variabel yang kita tulis.

Operator

Dalam PHP operator dibagi menjadi 4 yaitu operator *assignment*, operator aritmetika, operator relasional, dan operator logika. Operator *assignment* adalah operator yang digunakan untuk memberikan sebuah nilai yang akan dimasukkan ke dalam variabel. Operator *assignment* ditulis dengan karakter sama dengan ("="). Operator aritmetika digunakan untuk operasi aritmetika dengan daftar berikut.

| Operator | Keterangan |
|----------|--|
| + | Operator penjumlahan |
| - | Operator pengurangan |
| * | Operator perkalian |
| / | Operator pembagian |
| % | Operator modulus (sisanya dari pembagian) |
| ++ | Operator untuk menaikkan nilai variabel sebesar 1 |
| -- | Operator untuk menurunkan nilai variabel sebesar 1 |

Operator relasional adalah operator yang digunakan untuk membandingkan 2 buah nilai. Hasil operasi hanya ada dua yaitu *true* dan *false*. Berikut adalah contohnya.

| Operator | Keterangan |
|------------|--|
| \$a == \$b | Memeriksa apakah \$a sama dengan \$b |
| \$a != \$b | Memeriksa apakah \$a tidak sama dengan \$b |
| \$a < \$b | Memeriksa apakah \$a kurang dari \$b |
| \$a > \$b | Memeriksa apakah \$a lebih besar dari \$b |
| \$a <= \$b | Memeriksa apakah \$a kurang dari sama dengan \$b |
| \$a >= \$b | Memeriksa apakah \$a lebih dari sama dengan \$b |

Sedangkan operator logika adalah operator yang digunakan untuk mengoperasikan dua buah *operand* yang bertipe *boolean*. Contoh dari operator logika adalah sebagai berikut.

| Operator | Keterangan |
|----------|------------|
| && | Logika AND |
| | Logika OR |
| ! | Logika NOT |

c) Alur

Alur adalah sintaks yang didefinisikan untuk mengatur alur program. Pembahasan ini dibagi menjadi dua, yaitu:

- Kondisi
- Perulangan

Kondisi

Kondisi digunakan untuk menentukan kode mana yang akan dijalankan. Biasanya kondisi diawali dengan pengecekan beberapa parameter untuk diseleksi. Pada kondisi terdapat beberapa sintak yang bisa kita gunakan. Yang pertama adalah kondisi `if`. Berikut adalah contoh kodenya.

```
1  if (kondisi) {
2      statemen yang dieksekusi
3  }
```

Jika kita tidak menemukan hasil pada seleksi pertama maka kita bisa menambahkan `else` seperti contoh berikut.

```
1  if (kondisi) {
2      statemen yang dieksekusi
3  } else {
4      statemen lainnya
5  }
```

Atau kita bisa gunakan model seperti di bawah ini.

```
1  if (kondisi1){
2      // statemen untuk kondisi 1;
3  } elseif (kondisi2){
4      // statemen untuk kondisi 2;
5  } elseif (kondisi3){
6      // statemen untuk kondisi 3;
7  }
```

Selain `if-else`, kita juga bisa menggunakan kondisional `switch`, kondisional ini digunakan ketika kita membutuhkan banyak kondisi. Contohnya adalah sebagai berikut.

```
1  <?php
2      switch ($bulan){
3          case 1:
4              echo "Januari";
5              break;
6          case 2:
7              echo "Februari";
8              break;
9          case 3:
10             echo "Maret";
11             break;
12          case 4:
13             echo "April";
14             break;
15          case 5:
16             echo "Mei";
17             break;
18          case 6:
19             echo "Juni";
20             break;
21          case 7:
22             echo "Juli";
23             break;
24          case 8:
25             echo "Agustus";
26             break;
```

```

27     case 9:
28         echo "September";
29         break;
30     case 10:
31         echo "Oktober";
32         break;
33     case 11:
34         echo "November";
35         break;
36     case 12:
37         echo "Desember";
38         break;
39     default:
40         echo "Ini default statemen";
41     }
42     ?>

```

Perulangan

Dalam pembuatan sebuah aplikasi *web*, seringkali kita harus melakukan pengulangan terhadap statemen-statemen tertentu. PHP menyediakan kode yang dapat kita gunakan untuk mengulang statemen, antara lain *while*, *do-while*, *for*, dan *foreach*. Berikut sintaks masing-masing kode.

While

```

1 while(ekspresi) {
2     Statemen;
3 }

```

Do-while

```

1 do {
2     statemen
3 } while (ekspresi);

```

For

```

1 for (inisialisasi; ekspresi; increment/decrement) {
2     statemen
3 }

```

Foreach

```

1 foreach ($array as $current) {
2     statemen
3 }

```

4.4 Prosedur Praktikum

a) Latihan 1

Tulis kode di bawah ini.

```

1  <?php
2      $hello = "Hello World!";
3      $a_number = 4;
4      $anotherNumber = 8.9;
5      $bool = true;

```

Hapus notasi \$, lalu jalankan kode di atas. Apa yang terjadi? Lalu simpulkan.

b) Latihan 2

Tulis kode di bawah ini.

```

1  <?php
2      echo "<h1>Variabel</h1></br>";
3
4      $angka = 99;
5      echo "Ini adalah angka = $angka </br>";
6
7      $angka_2 = $angka + 1;
8      echo "Ini hasilnya = $angka_2 </br>";

```

Amati apa hasil dari kode di atas lalu simpulkan.

c) Latihan 3

Tulis kode di bawah ini.

```

1  <?php
2      $addition = 2 + 4;
3      $subtraction = 6 - 2;
4      $multiplication = 5 * 3;
5      $division = 15 / 3;
6      $modulus = 5 % 2;
7
8      echo "Penambahan: 2 + 4 = $addition <br>";
9      echo "Pengurangan: 6 - 2 = $subtraction <br>";
10     echo "Perkalian: 5 * 3 = $multiplication <br>";
11     echo "Pembagian: 15 / 3 = $division <br>";
12     echo "Pembagian sisa: 5 % 2 = $modulus";

```

Amati apa hasil dari kode di atas lalu simpulkan.

d) Latihan 4

Tulis kode di bawah ini.

| | |
|---|--|
| 1 | <?php |
| 2 | \$x = 4; |
| 3 | \$x += 3; |
| 4 | echo "Hasil dari operasi tersebut adalah = \$x"; |

Gantikan operator “+=” dengan:

1. “-=”
2. “*=”
3. “/=”
4. “%=”
5. “.=”

Amati apa hasil dari operator tersebut lalu simpulkan.

e) Latihan 5

Tulis kode di bawah ini.

| | |
|---|-----------------------------------|
| 1 | <?php |
| 2 | \$my_name = "anotherguy"; |
| 3 | |
| 4 | if (\$my_name == "someguy") { |
| 5 | echo "Your name is someguy! "; |
| 6 | } |
| 7 | |
| 8 | echo "Welcome to my homepage!"; |

Isikan variabel \$my_name dengan nilai yang lain dan simpulkan.

f) Latihan 6

Tulis kode di bawah ini.

```
1  <?php
2  $destination = "Tokyo";
3  echo "Traveling to $destination<br>";
4
5  switch ($destination){
6      case "Las Vegas":
7          echo "Bring an extra $500";
8          break;
9      case "Amsterdam":
10         echo "Bring an open mind";
11         break;
12     case "Egypt":
13         echo "Bring 15 bottles of SPF 50 Sunscreen";
14         break;
15     case "Tokyo":
16         echo "Bring lots of money";
17         break;
18     case "Caribbean Islands":
19         echo "Bring a swimsuit";
20         break;
21 }
```

Gantikan nilai dari variabel `$destination` dengan "Amsterdam". Apa yang terjadi? Jelaskan alur kerja dari alur `switch`.

g) Latihan 7

Tulis kode di bawah ini.

```
1  <?php
2  $brush_price = 5;
3
4  echo "<table border=\"1\" align=\"center\">";
5  echo "<tr><th>Quantity</th>";
6  echo "<th>Price</th></tr>";
7
8  for ( $counter = 10; $counter <= 100; $counter += 10) {
```

| | |
|----|---------------------------------|
| 9 | echo "<tr><td>"; |
| 10 | echo \$counter; |
| 11 | echo "</td><td>"; |
| 12 | echo \$brush_price * \$counter; |
| 13 | echo "</td></tr>"; |
| 14 | } |
| 15 | |
| 16 | echo "</table>"; |

Ganti inkremen dengan `$counter += 5`, apa yang terjadi? Ganti looping `for` dengan menggunakan `while`, dan `do-while`? Jelaskan bagaimana alur kerja dari looping `for`, `while`, dan `do-while`?
