



Topik

Pengenalan konsep dasar php

Tujuan

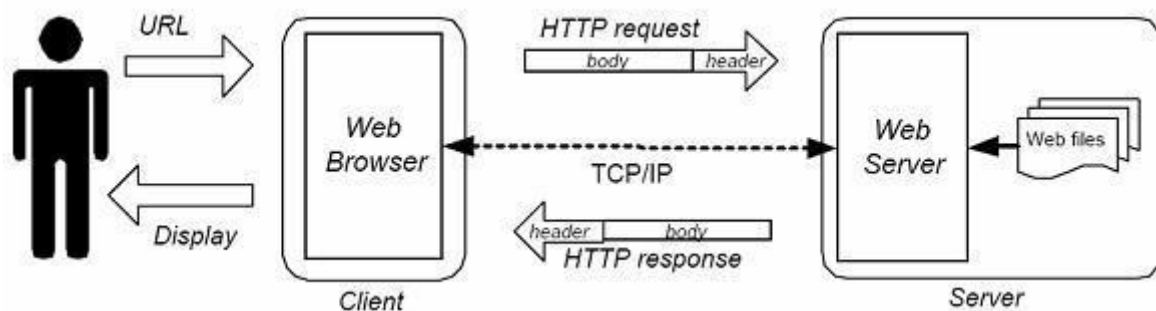
Mahasiswa diharapkan mampu:

1. memahami konsep dasar web server
2. memahami konsep php (*variable, type data, operator*)
3. mengimplementasikan konsep php ke dalam web dinamis

Pendahuluan

Pengenalan Web Server

Server atau web server adalah sebuah perangkat lunak (*software*) yang berfungsi untuk menerima permintaan (*request*) dari *client* (web browser) berupa halaman web melalui protokol HTTP/HTTPS, lalu merespon permintaan tersebut dalam bentuk halaman web berupa dokumen HTML. Dalam bahasa sederhananya web server diartikan sebagai *software* yang berfungsi sebagai pemberi layanan kepada web *client* (*browser*) seperti Chrome, Mozilla, Opera, Safari, dan lainnya, agar *browser* tersebut dapat menampilkan halaman website yang diminta oleh user. Salah satu contoh dari Web Server adalah Apache. Cara kerja web server diilustrasikan pada Gambar 1.



Gambar 1. Ilustrasi cara kerja web server (sumber gambar: google.com)

Seperti yang sudah disebutkan sebelumnya, *client* adalah pengguna internet yang memakai *browser* (Chrome, Mozilla, Opera, Safari, dan lainnya) untuk terhubung dengan web server melalui internet, sedangkan web server adalah *software* yang melayani permintaan web *client*. Ketika *client* (*browser*) melakukan permintaan data halaman website kepada server, maka permintaan tersebut dikemas oleh *browser* di dalam TCP (*Transmission Control Protocol*) yang merupakan protokol transport. Lalu permintaan tersebut dikirim ke alamat data, dalam hal ini merupakan protokol HTTP atau HTTPS. Selanjutnya, data tersebut akan dicari oleh web server di dalam komputer server. Jika data ditemukan maka data tersebut akan dikemas oleh web server dalam TCP lalu dikirim dan ditampilkan ke *browser*. Jika data yang dikirim diminta web *client* tidak ditemukan, maka web server akan menolak permintaan dan menampilkan halaman “Error 404” atau “Page Not Found” di *browser*.

Pengenalan PHP

Bahasa pemrograman PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf. Sejarah dan perkembangan bahasa pemrograman PHP dapat dibaca lebih lanjut pada situs resmi PHP yang beralamat di <http://www.php.net>. PHP berasal dari kata *Hypertext Preprocessor*. PHP disebut juga sebagai *server-side programming/script*, yaitu bahasa pemrograman yang akan dijalankan/diproses oleh web server. PHP sebenarnya bukan bahasa pemrograman yang wajib digunakan dalam membuat halaman web. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap. PHP digunakan untuk membuat website yang dinamis, dimana halaman web bisa menyesuaikan tampilan konten tergantung kondisi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari user, memproses form, dan sebagainya.

Untuk menjalankan kode program PHP, setidaknya dibutuhkan 3 jenis program, yaitu: web server, teks editor, dan web browser. Kode program PHP bisa diletakkan di dalam HTML atau dalam file yang terpisah. Agar web server dapat mengeksekusi kode program PHP, file harus disimpan sebagai file PHP, yaitu dengan akhiran file *.php. Sintak dasar kode program PHP adalah sebagai berikut:

```
<?php
    baris kode program
?>
```

Kode program PHP memiliki beberapa aturan dasar, yaitu:

1. *case sensitive*

PHP tidak membedakan huruf besar dan kecil untuk penamaan fungsi, class, maupun keyword bawaan PHP, seperti `echo` (perintah untuk menampilkan output ke layar), `while` (pengulangan), dan `if` (kondisi). Namun, PHP membedakan huruf besar dan huruf kecil (*case sensitive*) untuk penamaan variabel, sehingga variabel `$nama` dan `$NAMA` akan dianggap sebagai 2 variabel yang berbeda. Berikut contoh penulisan variable:

```
<?php
    $nama_pt = "polinema";
    echo $nama_pt; //output: polinema
    ECHO $nama_pt; //output: polinema
    echo $nama_PT; //output: error, karena tidak ada variabel $nama_PT
?>
```

2. Penulisan baris perintah dalam PHP

Baris perintah (*statement*) di dalam PHP adalah kumpulan perintah PHP yang menginstruksikan PHP untuk melakukan sesuatu. Setiap akhir baris perintah diakhiri dengan tanda *semicolon* atau titik koma (;). Kumpulan baris perintah yang menggunakan tanda kurung kurawal seperti kondisi (`if`) atau pengulangan (`loop`) tidak dibutuhkan tanda titik koma setelah kurung penutup.

```
<?php
    if(true){
        echo "perintah dijalankan";
    }
?>
```

3. Karakter spasi dan tab

Di dalam kode program PHP, karakter spasi dan tab diabaikan dalam eksekusi program.

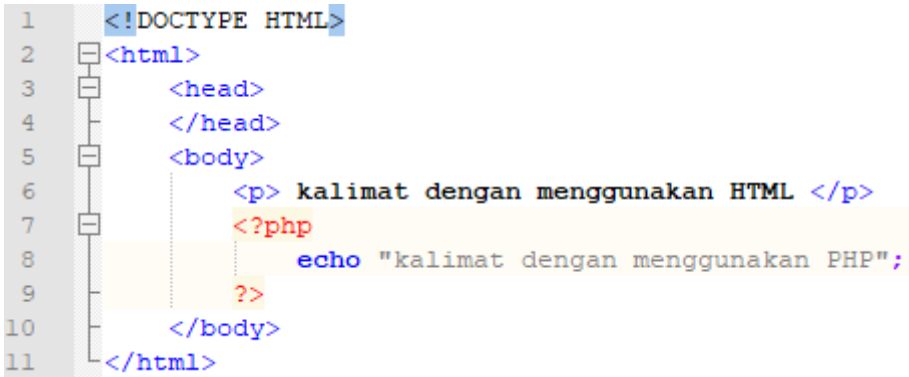
4. Penulisan komentar

PHP memberikan beberapa cara untuk menuliskan komentar, yaitu metode C/C++ dengan tanda (//) dan UNIX Shel dengan tanda (#). Berikut adalah contoh penulisan komentar:

```
<?php
    $luas = 1/2 * $alas * $tinggi; //mengitung luas segitiga
    echo $luas; #menampilkan hasil
?>
```

Praktikum Bagian 1. Menjalankan Kode Program PHP

Agar halaman web yang dimaksud dapat ditampilkan pada browser, diperlukan suatu cara agar web server dapat membedakan kode PHP dan HTML. Ikuti langkah-langkah berikut untuk memahami cara kerja web server menjalankan kode program PHP:

Langkah	Keterangan
1	Jalankan Apache pada XAMPP
2	Buatlah direktori baru bernama “praktik_php” di dalam direktori “dasarWeb” yang telah Anda buat pada praktikum sebelumnya.
3	Buat file baru dengan nama <code>cara_kerja.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut: 
4	Buat file baru dengan nama <code>cara_kerja.html</code> , kemudian salin kode program yang sama dengan langkah 2.
5	Simpan kedua file tersebut. Buka 2 tab/jendela browser dan jalankan kedua kode program tersebut dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cara_kerja.php</code> dan <code>localhost/dasarWeb/praktik_php/cara_kerja.html</code>
6	Amati hasil yang ditampilkan dan jelaskan dengan bahasa Anda sendiri cara kerja web server setelah menjalankan kedua kode program tersebut! (soal no 1) jika di file berekstensi <code>.html</code> tulisan <code>echo</code> tidak dapat berfungsi tapi jika di file berkstensi <code>.php</code> maka akan berfungsi

Praktikum Bagian 2. Echo dan Print

Di dalam PHP untuk menampilkan data ke layar digunakan perintah `echo` dan `print`. Ikuti langkah-langkah berikut untuk memahami penggunaan perintah `echo` dan `print`:

Langkah	Keterangan
1	<p>Buat file <code>cetak.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre>1 <!DOCTYPE html> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$txt1 = "Praktikum PHP"; 8 \$txt2 = "Variabel, Tipe Data, String"; 9 echo "<h2> \$txt1 </h2>"; 10 echo "Materi yang dipelajari " . \$txt2; 11 12 print "<h2> \$txt1 </h2>"; 13 print "Materi yang dipelajari " . \$txt2; 14 ?> 15 </body> 16 </html></pre>
2	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cetak.php</code></p>
3	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 2) fungsi <code>echo</code> dan <code>print</code> hampir sama tapi berbeda persamaannya sama menampilkan teks ke layar di fungsi <code>echo</code> akan mengembalikan nilai 1</p>

Praktikum Bagian 3. Variabel

Variable adalah kode program yang digunakan untuk menampung nilai tertentu. Nilai yang disimpan di dalam variable selanjutnya dapat dipindahkan ke dalam database atau ditampilkan kembali ke pengguna. Sebuah variable memiliki nama, yang digunakan untuk mengakses nilai dari variable itu. Nilai dari variabel dapat diisi dengan informasi yang diinginkan dan dapat diubah nilainya pada saat kode program sedang berjalan.

Berikut adalah aturan-aturan variable dalam PHP:

1. Penulisan variable diawali dengan tanda dollar (\$), kemudian diikuti dengan nama variabel.
2. Nama variable berisi karakter alfanumerik (A-z, 0-9, dan garis bawah atau *underscore*). Nama variabel harus diawali dengan huruf atau karakter garis bawah (_), tidak boleh diawali dengan angka.
3. Penulisan nama variable dalam PHP adalah *case sensitive*, sehingga perlu diperhatikan besar dan kecil huruf ketika menuliskannya.

4. Variable dalam PHP tidak perlu dideklarasikan terlebih dahulu

Ikuti langkah-langkah berikut untuk memahami penggunaan variabel PHP:

Langkah	Keterangan
1	<p>Buat file <code>variabel.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre>1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 hello = "Hello World!"; 8 echo \$hello; 9 ?> 10 </body> 11 </html></pre>
2	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/variabel.php</code></p>
3	<p>Modifikasi kode program pada langkah 1 dengan menambahkan tanda <code>\$</code> pada baris ke-7, sehingga kode program menjadi seperti berikut:</p> <pre>1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$hello = "Hello World!"; 8 echo \$hello; 9 ?> 10 </body> 11 </html></pre>
4	<p>Ulangi langkah ke-2</p>
5	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 3) di contoh pertama akan error karena nama variabel berbeda dan i file kedua akan menampilkan nilai variabel <code>\$hello</code></p>

Praktikum Bagian 4. Konstanta

Berbeda dengan variabel, konstanta berisikan nilai yang sifatnya tetap dan tidak bisa diubah sepanjang program dijalankan. Dalam PHP mendefinisikan konstanta menggunakan fungsi `define()`. Sintaks dasar konstanta adalah:

```
<?php
    define(name, value, case-insensitive)
?>
```

- Parameter `name`, adalah nama konstanta

- Parameter value, adalah nilai konstanta
- case-insensitive, berarti nama konstanta harus *case-insensitive*, default bernilai false

Ikuti langkah-langkah berikut untuk memahami penggunaan variabel dan konstanta PHP:

Langkah	Keterangan
1	<p>Buat file konstanta.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$var = "ini adalah contoh variabel"; 8 echo \$var; 9 echo "

"; 10 define("konstanta", "ini adalah contoh konstanta"); 11 echo konstanta; 12 echo "
"; 13 echo KONSTANTA; 14 <?> 15 </body> 16 </html> </pre>
2	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/konstanta.php. kemudian amati hasilnya</p>
3	<p>Modifikasi kode program langkah 1 dengan menambahkan parameter "true" di baris ke-10, sehingga kode program menjadi seperti berikut:</p> <pre> 1 <?DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$var = "ini adalah contoh variabel"; 8 echo \$var; 9 echo "

"; 10 define("konstanta", "ini adalah contoh konstanta", true); 11 echo konstanta; 12 echo "
"; 13 echo KONSTANTA; 14 <?> 15 </body> 16 </html> </pre>
4	<p>Ulangi langkah ke-2</p>
5	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 4) di percobaan pertama ada error dikarenakan php tidak setuju adanya variabel KONSTANTA dikarenakan berbeda maka di tambahkan paramameter true</p>

Praktikum Bagian 5. Tipe Data

Variabel dapat menyimpan data dengan tipe data yang berbeda. PHP mendukung beberapa tipe data, yaitu nilai skalar (integer, float, string, dan boolean), array, tipe data khusus (resource dan NULL). Berikut ini adalah penjelasan masing-masing tipe data:

Tipe Data	Keterangan
Integer	Integer adalah tipe data yang merepresentasikan bilangan bulat, positif maupun negatif, dan bukan pecahan. Rentang bilangan integer adalah -2,147,483,648 sampai 2,147,483,647.
Float	Float disebut juga sebagai bilangan pecahan atau sering juga disebut sebagai bilangan real yang mengandung angka decimal di belakang koma. Contoh bilangan float adalah 3,14.
String	String adalah tipe data karakter yang biasanya diekspresikan dengan diapit oleh tanda petik ganda (" ") atau petik tunggal (' '). Contoh string adalah "ini adalah string".
Boolean	Boolean adalah tipe data yang merepresentasikan nilai kebenaran (<i>truth value</i>). Sebaran nilai Boolean adalah <i>true</i> dan <i>false</i> .
Array	Array adalah variabel yang menyimpan sekelompok nilai, yang dapat diidentifikasi berdasarkan posisinya atau indeksinya.
Resource	Resource adalah tipe data yang menyimpan referensi resource eksternal.
NULL	NULL menunjukkan sebuah variabel belum memiliki nilai/data.

Dalam bahasa pemrograman PHP tidak seperti bahasa pemrograman lainnya, tidak perlu menuliskan tipe variabel di depan nama variabel. Di dalam PHP terdapat istilah *type juggling*, yaitu proses untuk menentukan jenis tipe data yang “cocok” untuk operasi saat itu dan PHP akan otomatis mengkonversinya. Ikuti langkah-langkah berikut untuk memahami tipe data dalam PHP:

Langkah	Keterangan
1	<p>Buat file <code>tipedata.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$a = "Hello World!"; 8 \$b = 99; 9 \$c = 8.8; 10 \$d = NULL; 11 var_dump (\$a,\$b,\$c,\$d); 12 ?> 13 </body> 14 </html> </pre>
2	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/tipedata.php</code></p>
3	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 5) di file ini akan print data + dengan nama typedta dilakukan oleh vardump()</p>

4	<p>Buat file <code>tipedata2.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 </head> 5 <body> 6 <h2> Konversi Tipe Data</h2> 7 <?php 8 \$a = 10.5; 9 \$b = "9 kucing"; 10 echo \$a; 11 echo "
".(integer) \$a; 12 echo "
". \$b; 13 echo "
". (integer) \$b; 14 ?> 15 </body> 16 </html> </pre>
5	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/tipedata2.php</code></p>
6	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 6)</p> <p>kita melakukan conversi tipe data \$a ke integer dan \$b juga sama</p>

Praktikum Bagian 6. Operator

Dalam PHP terdapat empat jenis operator, yaitu *assignment operator*, operator aritmatik, operator relasional, dan operator logika. *Assignment operator* adalah operator yang digunakan untuk memberikan sebuah nilai yang akan dimasukkan ke dalam variabel. *Assignment operator* adalah tanda sama dengan “=”. Operator aritmatik adalah operator yang digunakan dalam operasi aritmatika. Berikut ini adalah contoh operator aritmatik:

Operator	Keterangan
+	Operator penjumlahan
-	Operator pengurangan
*	Operator perkalian
/	Operator pembagian
%	Operator modulus (sis hasil pembangian)
++	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 contoh operator relasional:

Operator	Keterangan
<code>\$a == \$b</code>	Memeriksa apakah \$a sama dengan \$b

<code>\$a != \$b</code>	Memeriksa apakah \$a tidak sama dengan \$b
<code>\$a < \$b</code>	Memeriksa apakah \$a kurang dari \$b
<code>\$a > \$b</code>	Memeriksa apakah \$a lebih besar dari \$b
<code>\$a <= \$b</code>	Memeriksa apakah \$a kurang dari sama dengan \$b
<code>\$a >= \$b</code>	Memeriksa apakah \$a lebih dari sama dengan \$b

Operator logika adalah operator yang digunakan untuk mengoperasikan dua buah operand yang bertipe Boolean. Contoh dari operator logika adalah:

Operator	Keterangan
<code>\$a and \$b</code>	Bernilai benar jika \$a dan \$b keduanya benar
<code>\$a or \$b</code>	Bernilai benar jika \$a atau \$b bernilai benar
<code>\$a xor \$b</code>	Bernilai benar jika \$a atau \$b bernilai benar, tetapi tidak keduanya
<code>\$a && \$b</code>	Logika AND
<code>\$a \$b</code>	Logika OR
<code>!\$a</code>	Logika NOT

Ikuti langkah-langkah berikut untuk lebih memahami operator di dalam PHP:

Langkah	Keterangan
1	<p>Buat file <code>operator.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$a =5; 8 \$b =2; 9 echo "hasil penambahan \$a dan \$b adalah "; echo \$a + \$b; 10 echo "
 hasil pengurangan \$a dan \$b adalah "; echo \$a - \$b; 11 echo "
 hasil perkalian \$a dan \$b adalah "; echo \$a * \$b; 12 echo "
 hasil pembagian \$a dan \$b adalah "; echo \$a / \$b; 13 echo "
 hasil sisa pembagian \$a dan \$b adalah "; echo \$a % \$b; 14 ?> 15 </body> 16 </html> </pre>
2	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/operator.php</code></p>
3	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 7) tampilan sesuai dengan operator matematika</p>
4	<p>Buat file <code>operator_2.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p>

	<pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$a = 12; 8 \$b = 14; 9 echo "\$a < \$b adalah "; var_dump(\$a < \$b); 10 echo "
 \$a > \$b adalah "; var_dump(\$a > \$b); 11 echo "
 \$a == \$b adalah "; var_dump(\$a == \$b); 12 echo "
 \$a != \$b adalah "; var_dump(\$a != \$b); 13 ?> 14 </body> 15 </html> </pre>
5	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/operator_2.php</p>
6	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 8) tampilan sesuai dengan operator aritmatika</p>
7	<p>Buat file operator_3.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$a=true; 8 \$b=false; 9 10 echo "nilai a AND b adalah "; var_dump(\$a and \$b); 11 echo "
 nilai a OR b adalah "; var_dump(\$a or \$b); 12 echo "
 nilai a XOR b adalah "; var_dump(\$a xor \$b); 13 echo "

 nilai a && b adalah "; var_dump(\$a && \$b); 14 echo "
 nilai a b adalah "; var_dump(\$a \$b); 15 echo "
 nilai !a b adalah "; var_dump(!\$a \$b); 16 ?> 17 </body> 18 </html> </pre>
8	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/operator_3.php</p>
9	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 9) tampilan sesuai dengan operator logika</p>

10	<p>Buat file <code>operator_4.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$x =4; 8 \$x += 3; 9 echo "hasil operasi tersebut adalah = \$x"; 10 ?> 11 </body> 12 </html> </pre>
11	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/operator_4.php</code></p>
12	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 10) hasil nya 7 karena 4+3</p>
13	<p>Modifikasi kode program langkah ke-4, ganti operator “+=” pada baris ke-8 dengan operator-operator berikut ini:</p> <ol style="list-style-type: none"> “-= ” “*= ” “/= ” “%= ” “.= ”
14	<p>Amati hasil dari operator-operator pada langkah ke-12, lalu simpulkan hasil pengamatanmu (soal no 11)</p> <p>hasil nya 1 karena 4-3 hasil nya 1.3 karena 4/3 hasil nya 1 karena sisa nya 3/4 hasil nya 3 karena 3 diinisialisasi terakhir</p>
15	<p>Buat file <code>operator_5.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p>

	<pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$nomor = 1; 8 while(\$nomor <= 5) { 9 echo \$nomor++; 10 } 11 ?> 12 </body> 13 </html> </pre>
16	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/operator_5.php</p>
17	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 12) akan melakukan perulangan while dari 1 sampai 5</p>
18	<p>Modifikasi kode program pada langkah ke-14 dengan mengubah baris ke-9 dengan pre-increment, sehingga kode program menjadi seperti berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <?php 7 \$nomor = 1; 8 while(\$nomor <= 5) { 9 echo ++\$nomor; 10 } 11 ?> 12 </body> 13 </html> </pre>
19	<p>Simpan file tersebut dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/operator_5.php</p>
20	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 13) akan melakukan perulangan while dari 2 sampai 6 karena nilai sudah memiliki nilai yaitu 1</p>

Referensi :

- 1) Nixon, Robin. (2018). Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-step Guide to Creating Dynamic Websites, 5th Edition. O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). The Joy of PHP: A Beginners's Guide to Programming Interactive Web Applications with PHP and MySQL, 5th Edition. Plum Island Publishing.