

Assignment & Operator Aritmatika

Sub CPMK:

Setelah mengikuti modul ini, mahasiswa diharapkan mampu:

- Memahami dan mengimplentasikan operator assignment & operator aritmatika di dalam PHP

Operator Assignment

- Operator assignment → operator untuk menambahkan atau memasukkan sebuah nilai kedalam variabel.
- PHP memiliki 3 jenis operator assignment:
 1. Assignment dengan Nilai (Assignment by Value)
 2. Assignment Array
 3. Assignment dengan Referensi (Assignment by Reference)

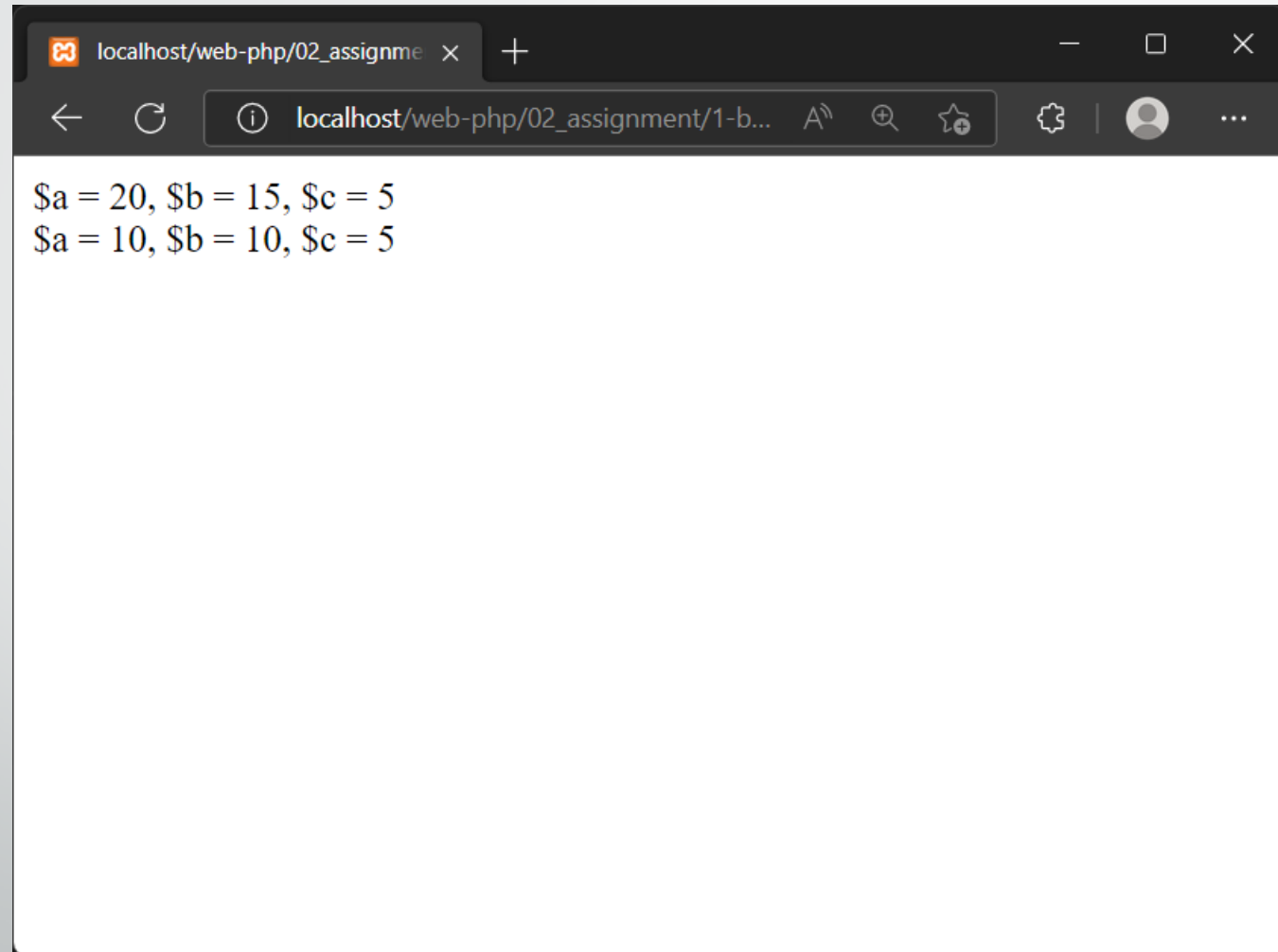
1. Assignment dengan Nilai (Assignment by Value)

- Assignment dengan Nilai atau dalam istilah pemrograman dikenal dengan Assignment by Value → proses pemberian nilai kedalam sebuah variabel dengan meng-copy nilai atau value dari variabel lain.
- PHP menggunakan tanda sama dengan (=) untuk Assignment by Value. Pada saat menggunakan operator "=", PHP meng-copy nilai (atau value), dari sisi kanan operator ke sisi kiri.

```
1-byvalue.php X
02_assignment > 1-byvalue.php
1  <?php
2  $a = 20;
3  $b = 15;
4  $c = 5;
5
6  echo "\$a = $a, \$b = $b, \$c = $c";
7  echo "<br />";
8  // hasil proses: $a = 20, $b = 15, $c = 5
9
10 $a = $b = $c+5;
11 echo "\$a = $a, \$b = $b, \$c = $c";
12 // hasil proses: $a = 10, $b = 10, $c = 5
13 ?>
```

- Pada contoh kode PHP diatas, kita defenisikan 3 variabel: \$a, \$b, dan \$c dengan operator assignment by value. Perhatikan pada baris terakhir (11), dimana kode \$a = \$b = \$c+5, yang urutan pemrosesannya di mulai dari kanan ke kiri, sehingga yang diproses oleh PHP menjadi: \$a = (\$b = (\$c+5))

Hasil:



A screenshot of a web browser window. The address bar shows the URL `localhost/web-php/02_assignment/1-b...`. The page content displays two lines of PHP output: `$a = 20, $b = 15, $c = 5` followed by `$a = 10, $b = 10, $c = 5` on the next line.

```
$a = 20, $b = 15, $c = 5  
$a = 10, $b = 10, $c = 5
```

2. Assignment Array

- Assignment array → operator assignment untuk menginput nilai kedalam array. Operator ini menggunakan tanda panah (=>).
- Array (atau larik dalam bahasa indonesia) bukanlah tipe data dasar seperti integer atau boolean, Array adalah sebuah tipe data bentukan yang terdiri dari kumpulan tipe data lainnya. Menggunakan array akan memudahkan dalam membuat kelompok data, serta menghemat penulisan dan penggunaan variable.
- Cara penulisan array:

```
$nama_variabel = array(  
    key => value,  
    key2 => value2,  
    key3 => value3,  
    ...  
)
```

2. Assignment Array

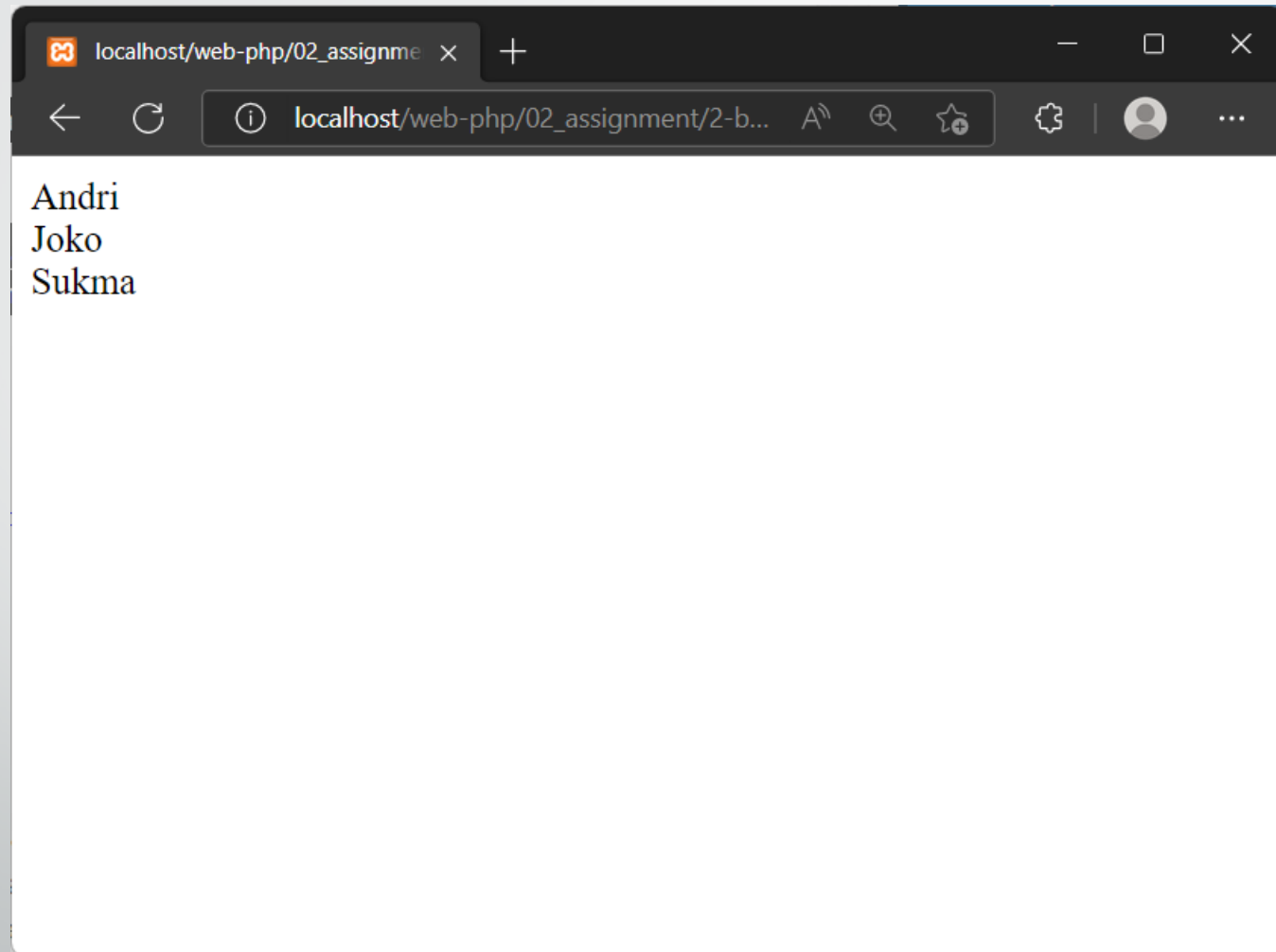
- Komponen array terdiri dari pasangan kunci (key) dan nilai (value). Key adalah penunjuk posisi dimana value disimpan. Perhatikan juga bahwa PHP menggunakan tanda panah (=>) untuk memberikan nilai kepada key.
- Dalam mengakses nilai dari array, kita menggunakan kombinasi \$nama_variabel dan nilai key-nya, dengan penulisan sebagai berikut:

```
$nama_variabel[key];
```


- Dalam contoh disamping, kita menggunakan angka integer sebagai key (1,2,3...) dan string sebagai value (Andri, Joko, Sukma, ...).
- Selain mendefenisikan key secara langsung, PHP juga memperbolehkan penulisan array tanpa key dan key itu secara otomatis akan diurutkan dari nilai 0, 1, 2, dst

```
2-byarray1.php X
02_assignment > 2-byarray1.php
1  <?php
2  //pembuatan array
3  $nama = array(
4      1=>"Andri",
5      2=>"Joko",
6      3=>"Sukma",
7      4=>"Rina",
8      5=>"Sari");
9
10 //cara akses array
11 echo $nama[1]; //Andri
12 echo "<br />";
13 echo $nama[2]; //Joko
14 echo "<br />";
15 echo $nama[3]; //Sukma
16 ?>
```

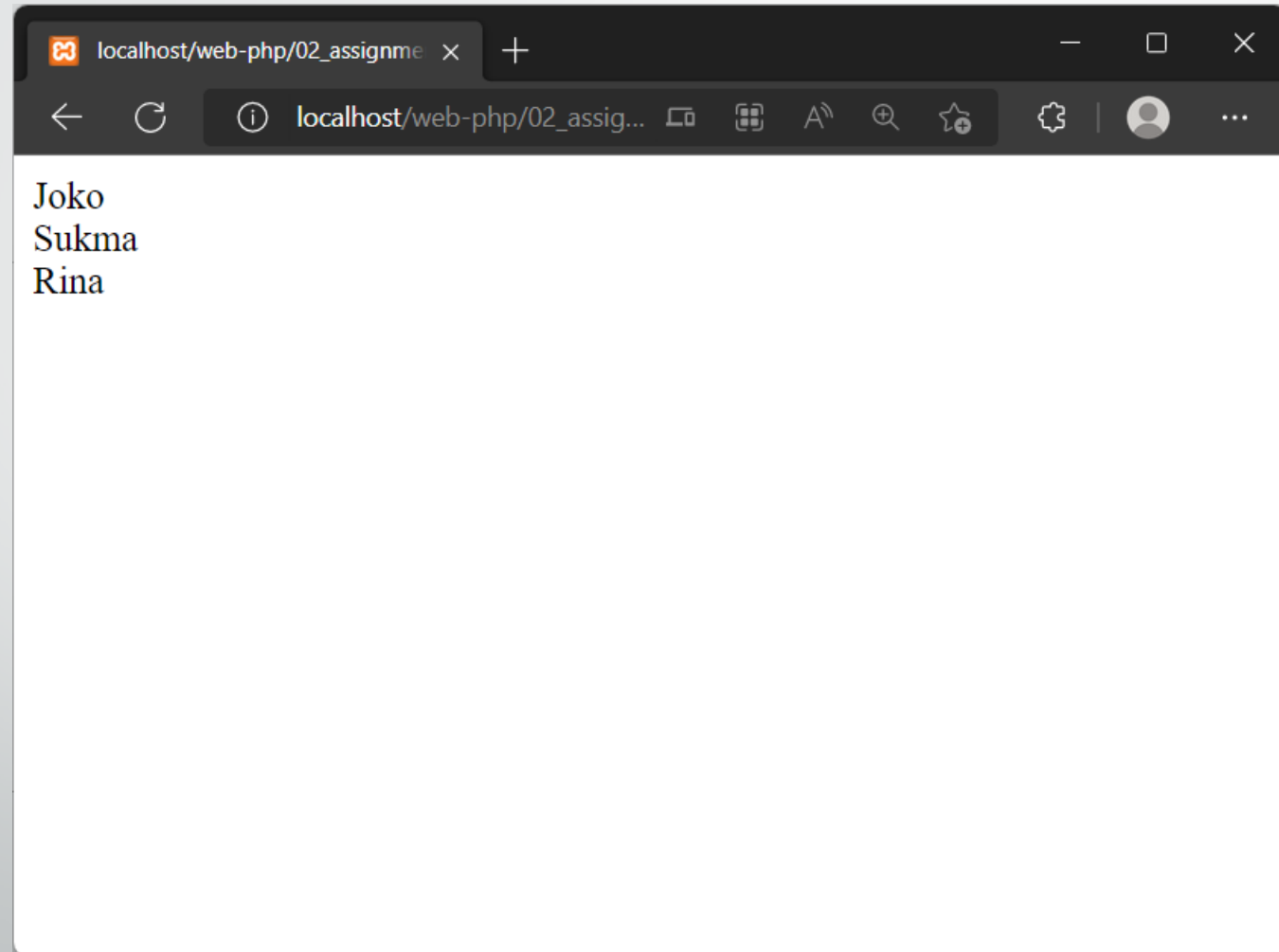
Hasil:



```
2-byarray2.php X
02_assignment > 2-byarray2.php
1  <?php
2      // pembuatan array
3      $nama = ["Andri","Joko","Sukma","Rina","Sari"];
4
5      // pengaksesan array
6      echo $nama[1]; //Joko
7      echo "<br />";
8      echo $nama[2]; //Sukma
9      echo "<br />";
10     echo $nama[3]; //Rina
11     ?>
```

- Perhatikan bahwa kita tidak perlu membuat keyword 'array', tapi langsung membuat tanda kurung siku.
- Seperti biasa, index key array dimulai dari 0. Sehingga pemanggilan \$nama[1] menghasilkan "Joko", bukan "Andri".

Hasil:



3. Assignment dengan Referensi (Assignment by Reference)

- Assignment dengan referensi atau dalam istilah programmingnya: Assignment by Reference → operator assignment khusus yang digunakan untuk men-copy nilai referensi dari sebuah variabel.

- Dalam kode program disamping, kita membuat 2 buah variabel, yaitu \$a dan \$b. Variabel \$a diinput dengan nilai 20, sedangkan variabel \$b men-copy nilai dari variabel \$a. Selanjutnya kita tampilkan kedua variabel tersebut menggunakan perintah echo.
- Pada baris ke-9 kita menambahkan nilai variabel \$a dengan 5, lalu menampilkan hasil kedua variabel tersebut. Selanjutnya pada baris ke-14 kita menambahkan \$b dengan 10, lalu menampilkan hasilnya
- Kata kunci disini adalah, variabel \$b hanya men-copy nilai yang ada pada variabel \$a, sehingga kedua variabel memiliki nilai sendiri-sendiri dan terpisah, seperti yang terlihat dari hasil echo.

```
3-byreference1.php X
02_assignment > 3-byreference1.php
1  <?php
2  $a = 20;
3  $b = $a;
4
5  echo "\$a = $a, \$b = $b";
6  echo "<br />";
7  // hasil proses: $a = 20, $b = 20
8
9  $a = $a + 5;
10 echo "\$a = $a, \$b = $b";
11 echo "<br />";
12 // hasil proses: $a = 25, $b = 20
13
14 $b = $b + 10;
15 echo "\$a = $a, \$b = $b";
16 // hasil proses: $a = 25, $b = 30
17 ?>
```

- Bagaimana jika yang kita inginkan adalah: karena variabel `$a=$b`, maka ketika kita merubah nilai salah satu variabel, nilai pada variabel yang lain juga ikut berubah. Fitur inilah yang bisa didapatkan dengan Assignment by Reference.
- Assignment by Reference dalam PHP menggunakan operator `"= &"`.
- Seperti yang dapat dilihat, bahwa sekarang kedua variabel (`$a` dan `$b`), seolah-olah saling terikat, sehingga ketika sebuah variabel diubah nilainya, variabel yang lain juga ikut berubah.

🐘 3-byreference2.php X

02_assignment > 🐘 3-byreference2.php

```
1  <?php
2  $a = 20;
3  $b = &$a;
4
5  echo "\$a = $a, \$b = $b";
6  echo "<br />";
7  // hasil proses: $a = 20, $b = 20
8
9  $a = $a + 5;
10 echo "\$a = $a, \$b = $b";
11 echo "<br />";
12 // hasil proses: $a = 25, $b = 25
13
14 $b = $b + 10;
15 echo "\$a = $a, \$b = $b";
16 // hasil proses: $a = 35, $b = 35
17 ?>
```

Operator Aritmatik

1. Operator Aritmatika → operator matematis yang terdiri dari operator penambahan, pengurangan, perkalian, pembagian, modulus, plus, dan minus

Operator	Makna	Contoh
+	Penjumlahan	$2 + 4$
-	Pengurangan	$6 - 2$
*	Perkalian	$5 * 3$
/	Pembagian	$15 / 3$
%	Modulus/sisa hasil bagi	$43 \% 10$

- $5 \% 2 = 1$, karena nilai sisa dari pembagian $5 / 2$ adalah 1.
- $5 = (2 * 2) + 1$

4-aritmatik.php

02_assignment > 4-aritmatik.php

```
1  <?php
2  $penjumlahan = 2 + 4;
3  $pengurangan = 6 - 2;
4  $perkalian = 5 * 3;
5  $pembagian = 15 / 3;
6  $modulus = 5 % 2;
7  echo "Hasil: 2 + 4 = " . $penjumlahan."<br>";
8  // Hasil: 2 + 4 = 6
9  echo "Hasil: 6 - 2 = " . $pengurangan."<br>";
10 // Hasil: 2 + 4 = 6
11 echo "Hasil: 5 * 3 = " . $perkalian."<br>";
12 // Hasil: 5 * 3 = 15
13 echo "Hasil: 15 / 3 = " . $pembagian."<br>";
14 // Hasil: 5 * 3 = 15
15 echo "Hasil: 5 % 2 = " . $modulus;
16 // Hasil: 5 % 2 = 1
17 ?>
```

Operator Aritmatik

2. Tingkat presedensi → mengerjakan terlebih dahulu operasi matematika yang lebih tinggi derajatnya

- Operator (/) memiliki tingkat presedensi yang sama dengan (*).
- Keduanya memiliki tingkat presedensi yang lebih tinggi daripada + dan -.
- Sedangkan operator modulo (%) levelnya juga sama dengan * dan /.

5-presedensi.php X

02_assignment > 5-presedensi.php

```
1  <?php
2  $a = 3 + 4 * 5 - 6;
3  echo $a;
4  // hasil $a = 17
5  echo "<br />";
6  $a = (3 + 4) * 5 - 6;
7  echo $a;
8  // hasil $a = 29
9  ?>
```

Operator Aritmatik

3. Pre/Post Increment dan Decrement
→ Operator ini digunakan pada proses menambah maupun mengurangi dengan tingkat 1

- $\$x++$; → ekuivalen dengan $\rightarrow \$x += 1$; atau $\$x = \$x + 1$;
- $\$x--$; → ekuivalen dengan $\rightarrow \$x -= 1$; atau $\$x = \$x - 1$;

```
6-increment.php X
02_assignment > 6-increment.php
1  <?php
2  $x = 4;
3  $x++;
4  echo "Nilai x yang baru : ". $x;
5  echo "<br />";
6  // hasil $x = 5
7  $x = 4;
8  $x--;
9  echo "Nilai x yang baru : ". $x;
10 // hasil $x = 3
11 ?>
```



Contoh studi kasus

script5-1.php X

02_assignment > script5-1.php

```
1  <html>
2      <head>
3          <title>Menghitung Komisi Salesman</title>
4      </head>
5      <body>
6          <h1>Menghitung Komisi Salesman</h1>
7          <?php
8              /*
9              Script ini akan menghitung komisi salesman berdasarkan nilai penjualan
10             yang dicapainya yaitu sebesar Rp. 1.500.000,-
11             Ketentuan komisinya adalah 5% dari nilai penjualan yang dicapai.
12             */
13             $nilaiJual = 1500000; // nilai penjualan yang didapat salesman
14             $komisi = 0.05 * $nilaiJual; // menghitung komisi yaitu 5% dari nilai penjualan
15             echo "<p>Nilai penjualan salesman : Rp. ".$nilaiJual."</p>"; // menampilkan nilai penjualan salesman
16             echo "<p>Komisi yang didapat salesman adalah Rp. ".$komisi."</p>";
17             // menampilkan hasil perhitungan komisi
18             ?>
19         </body>
20 </html>
```

script5-2.php X

02_assignment > script5-2.php

```
1  <html>
2      <head>
3          <title>Menghitung Gaji Bersih Karyawan</title>
4      </head>
5      <body>
6          <h1>Menghitung Gaji Bersih Karyawan</h1>
7          <?php
8              /*
9               Script ini akan menghitung gaji bersih karyawan yang dirumuskan dengan
10              Gaji Bersih = Gaji Pokok + tunjangan - pajak;
11              Misalkan gaji pokoknya Rp. 1.000.000, tunjangan Rp. 500.000 dan
12              pajaknya 15% dari (gaji kotor = gaji pokok + tunjangan)
13              Berikut ini ada beberapa cara pembuatan script yang akan menghasilkan
14              output yang sama
15              */
16              // CARA KE - 1
17              $gajiPokok = 1000000; // gaji pokok
18              $tunjangan = 500000; // tunjangan
19              $gajiKotor = $gajiPokok + $tunjangan; // hitung gaji kotor
20              $pajak = 0.15 * $gajiKotor; // hitung pajak
```

script5-2.php X

02_assignment > script5-2.php

```
20      $pajak = 0.15 * $gajiKotor; // hitung pajak
21      $gajiBersih = $gajiPokok + $tunjangan - $pajak; // hitung gaji bersih
22      echo "<p>Gaji bersih karyawan adalah Rp. ".$gajiBersih."</p>"; // menampilkan gaji bersih
23      // CARA KE - 2
24      $gajiPokok = 1000000; // gaji pokok
25      $tunjangan = 500000; // tunjangan
26      $gajiKotor = $gajiPokok + $tunjangan; // hitung gaji kotor
27      $gajiBersih = $gajiKotor - (0.15 * $gajiKotor); // hitung gaji bersih
28      echo "<p>Gaji bersih karyawan adalah Rp. ".$gajiBersih."</p>"; // menampilkan gaji bersih
29      // CARA KE - 3
30      $gajiPokok = 1000000; // gaji pokok
31      $tunjangan = 500000; // tunjangan
32      $gajiBersih = $gajiPokok + $tunjangan - 0.15 * ($gajiPokok +
33      $tunjangan); // hitung gaji bersih
34      echo "<p>Gaji bersih karyawan adalah Rp. ".$gajiBersih."</p>"; // menampilkan gaji bersih
35      ?>
36      </body>
37      </html>
```

script5-3.php X

02_assignment > script5-3.php

```
1  <html>
2      <head>
3          <title>Konversi Waktu Tempuh Ke Detik</title>
4      </head>
5      <body>
6          <h1>Konversi Waktu Tempuh Ke Detik</h1>
7          <?php
8              /*
9              Script ini akan mengkonversi waktu yang dinyatakan dalam 10:16:42 (10
10             jam, 16 menit dan 42 detik) ke dalam satuan detik.
11             */
12             $jam = 10;
13             $menit = 16;
14             $detik = 42;
15             $jamKeDetik = $jam * 3600; // konversi jam ke detik
16             $menitKeDetik = $menit * 60; // konversi menit ke detik
17             $detikKeDetik = $detik; // konversi ke detik
18             $totalDetik = $jamKeDetik + $menitKeDetik + $detikKeDetik; // hitung total waktu dalam detik
19             echo "<p>Jika waktu ".$jam." : ".$menit." : ".$detik." dinyatakan dalam
20             satuan detik adalah : ".$totalDetik."</p>";
21             ?>
22      </body>
23 </html>
```


script5-4.php X

02_assignment > script5-4.php

```
1  <html>
2      <head>
3          <title>Konversi jumlah detik ke satuan jam-menit-detik </title>
4      </head>
5      <body>
6          <h1>Konversi jumlah detik ke satuan jam-menit-detik</h1>
7          <?php
8              /*
9              Script ini merupakan kebalikan dari script5-3.php
10             Script ini akan mengkonversi waktu yang diketahui dalam satuan detik
11             ke dalam satuan jam-menit-detik.
12             Diketahui waktu dalam detik adalah 15789 detik, akan dikonversi ke
13             bentuk x jam, y menit dan z detik
14             */
15             $totalDetik = 15789; // jumlah total detik mula-mula
```

script5-4.php X

02_assignment > script5-4.php

```
16      // mencari waktu dalam jam
17      $sisas = $totalDetik % 3600;
18      $dalamJam = ($totalDetik - $sisas) / 3600;
19      // sisa dari perhitungan jam digunakan untuk menghitung menitnya
20      $totalDetik = $sisas;
21      $sisas = $totalDetik % 60;
22      $dalamMenit = ($totalDetik - $sisas) / 60;
23      // sisa dalam perhitungan menit digunakan untuk menghitung detikanya
24      $totalDetik = $sisas;
25      $sisas = $totalDetik % 1;
26      $dalamDetik = ($totalDetik - $sisas) / 1;
27      echo "<p>Hasil konversinya adalah : ".$dalamJam." jam :
28      ".$dalamMenit." menit : ".$dalamDetik." detik</p>";
29      ?>
30  </body>
31  </html>
```

script5-5.php X

02_assignment > script5-5.php

```
1  <html>
2      <head>
3          <title>Menghitung selisih dua buah waktu</title>
4      </head>
5      <body>
6          <h1>Menghitung selisih dua buah waktu</h1>
7          <?php
8              /*
9              Script ini akan mencari selisih antara waktu 10:34:45 dengan 12:25:31.
10             Hasil selisih waktu dinyatakan dalam detik
11             */
12             $jam1 = 12;
13             $menit1 = 25;
14             $detik1 = 31;
15             $jam2 = 10;
16             $menit2 = 34;
17             $detik2 = 45;
18             $totalDetik1 = $jam1 * 3600 + $menit1 * 60 + $detik1; // menghitung total detik untuk waktu pertama
19             $totalDetik2 = $jam2 * 3600 + $menit2 * 60 + $detik2; // menghitung total detik untuk waktu kedua
20             $selisih = $totalDetik1 - $totalDetik2; // hitung selisih total detik dari kedua waktu
21             echo "<p>Selisih dari kedua waktu adalah ".$selisih." detik</p>";
22         ?>
23     </body>
24 </html>
```