**LATIHAN 1**

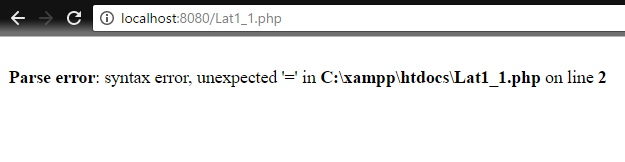
1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_1.php !

|  |  |
| --- | --- |
|  | **<?php**  $hello = "Hello World!";  $a\_number = 4;  $anotherNumber = 8.9;  $bool = true;  **?>** |

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

1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **<?php**  hello = "Hello World!";  a\_number = 4;  anotherNumber = 8.9;  bool = true;  **?>** |

1. **Penjelasan**

Notasi ‘$’ digunakan untuk mendeklarasikan variabel, maka apabila tanda tersebut dihilangkan, keyword tidak akan dikenali dan menghasilkan error

**LATIHAN 2**

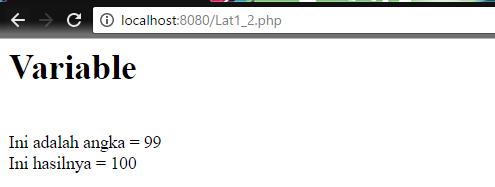
1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_2.php !

|  |  |
| --- | --- |
|  | **<?php**  echo "<h1>Variable</h1></br>";  $angka = 99;  echo "Ini adalah angka = $angka </br>";  $angka\_2 = $angka + 1;  echo "Ini hasilnya = $angka\_2 </br>";  **?>** |

Amati apa hasil dari kode di atas ! Lalu simpulkan.

1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **<?php**  echo "<h1>Variable</h1></br>";  $angka = 99;  echo "Ini adalah angka = $angka </br>";  $angka\_2 = $angka + 1;  echo "Ini hasilnya = $angka\_2 </br>";  **?>** |

1. **Penjelasan**

Kode diatas berfungsi untuk mencetak header dan nilai dari variabel angka dengan perintah echo. Maka akan muncul seperti berikut :  
Variable  
Ini adalah angka = 99  
Ini hasilnya = 100

**LATIHAN 3**

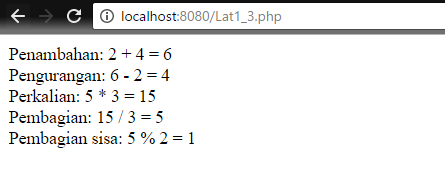
1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_3.php !

|  |  |
| --- | --- |
|  | **<?php**  $addition = 2 + 4;  $subtraction = 6 - 2;  $multiplication = 5 \* 3;  $division = 15 / 3;  $modulus = 5 % 2;  echo "Penambahan: 2 + 4 = $addition <br />";  echo "Pengurangan: 6 - 2 = $subtraction <br />";  echo "Perkalian: 5 \* 3 = $multiplication <br />";  echo "Pembagian: 15 / 3 = $division <br />";  echo "Pembagian sisa: 5 % 2 = $modulus";  **?>** |

Amati apa hasil dari kode di atas ! Lalu simpulkan.

1. **Screenshoot**



1. **Syntax**

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

1. **Penjelasan**

Kode diatas akan mengolah perhitungan yang ada dan dimasukkan kedalam variabel-variabel lalu di tampilkan hasilnya dengan perintah echo.

dan menghasilkan :

Penambagan: 2 + 4 = 6

Pengurangan: 6 – 2 = 4  
dan seterusnya

**LATIHAN 4**

1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_4.php !

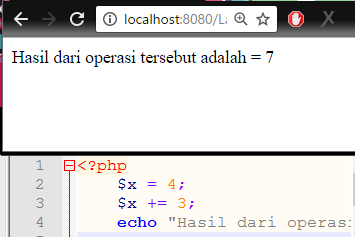
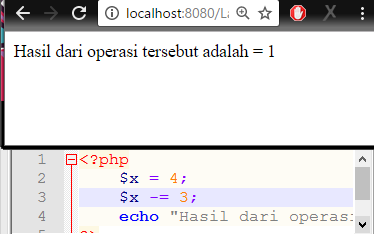
|  |  |
| --- | --- |
|  | **<?php**  $x = 4;  $x += 3;  echo "Hasil dari operasi tersebut adalah = $x";  **?>** |

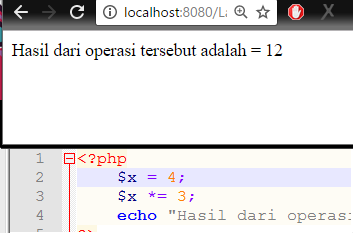
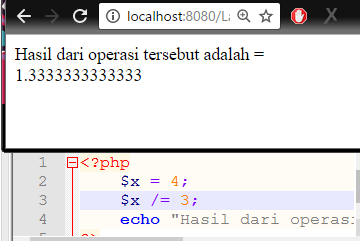
Gantikan operator “+=” dengan :

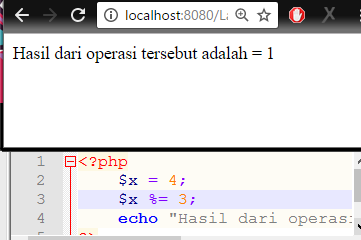
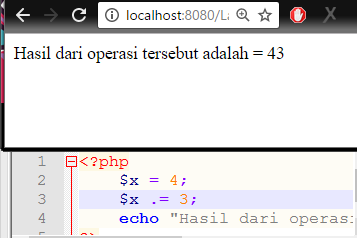
1. “-=”
2. “\*=”
3. “/=”
4. “%=”
5. “.=”

Amati apa hasil dari operator tersebut ! lalu simpulkan.

1. **Screenshoot**

1. **Syntax**

|  |  |
| --- | --- |
| 1  2  3  4  5 | **<?php**  **$x = 4;**  **$x += 3;**  **echo "Hasil dari operasi tersebut adalah = $x";**  **?>** |
| 1  2  3  4  5 | **<?php**  **$x = 4;**  **$x -= 3;**  **echo "Hasil dari operasi tersebut adalah = $x";**  **?>** |
| 1  2  3  4  5 | **<?php**  **$x = 4;**  **$x \*= 3;**  **echo "Hasil dari operasi tersebut adalah = $x";**  **?>** |
| 1  2  3  4  5 | **<?php**  **$x = 4;**  **$x /= 3;**  **echo "Hasil dari operasi tersebut adalah = $x";**  **?>** |
| 1  2  3  4  5 | **<?php**  **$x = 4;**  **$x %= 3;**  **echo "Hasil dari operasi tersebut adalah = $x";**  **?>** |
| 1  2  3  4  5 | **<?php**  **$x = 4;**  **$x .= 3;**  **echo "Hasil dari operasi tersebut adalah = $x";**  **?>** |

1. **Penjelasan**

Setiap operator perhitungan akan melakukan oprasi berdasarkan isi dari variabel yang sudah ada.

Operator += akan menjalankan operator penjumlahan dan hasilnya disimpan ke variabel x, operator -= adalah operator pengurangan, operator /= adalah operator pembagian, operator %= adalah operator modulus/sisa pembagian, operator .= adalah operator concat yang akan menggabungkan 2 string menjadi satu. Lalu akan ditampilkan hasil perhitungannya melalui perintah echo.

**LATIHAN 5**

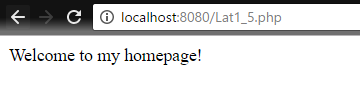
1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_5.php !

|  |  |
| --- | --- |
|  | **<?php**  $my\_name = "anotherguy";  if ( $my\_name == "someguy" ) {  echo "Your name is someguy!<br />";  }  echo "Welcome to my homepage!";  **?>** |

Isikan variable $my\_name dengan nilai yang lain dan simpulkan!

1. **Screenshoot**



1. **Syntax**

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

1. **Penjelasan**

Ketika Variabel my\_name diisi dengan nilai yang lain , maka program tidak akan memunculkan “Yourname is someguy!” melainkan “Welcome to my homepage!”

**LATIHAN 6**

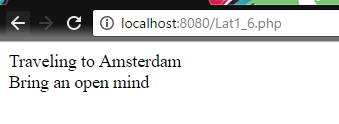
1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_6.php

|  |  |
| --- | --- |
|  | **<?php**  $destination = "Tokyo";  echo "Traveling to $destination<br />";  switch ($destination){  case "Las Vegas":  echo "Bring an extra $500";  break;  case "Amsterdam":  echo "Bring an open mind";  break;  case "Egypt":  echo "Bring 15 bottles of SPF 50 Sunscreen";  break;  case "Tokyo":  echo "Bring lots of money";  break;  case "Caribbean Islands":  echo "Bring a swimsuit";  break;  }  **?>** |

Gantikan nilai dari variable $destination dengan “Amsterdam”, apa yang terjadi? Jelaskan alur kerja dari model switch!

1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | **<?php**  $destination = "Tokyo";  echo "Traveling to $destination<br />";  switch ($destination){  case "Las Vegas":  echo "Bring an extra $500";  break;  case "Amsterdam":  echo "Bring an open mind";  break;  case "Egypt":  echo "Bring 15 bottles of SPF 50 Sunscreen";  break;  case "Tokyo":  echo "Bring lots of money";  break;  case "Caribbean Islands":  echo "Bring a swimsuit";  break;  }  **?>** |

1. **Penjelasan**

Ketika nilai $destionation diubah dengan amsterdam maka program akan langsung memanggil Case amsterdam karena nilai di $Destination sesuai dengan case tersebut.

**LATIHAN 7**

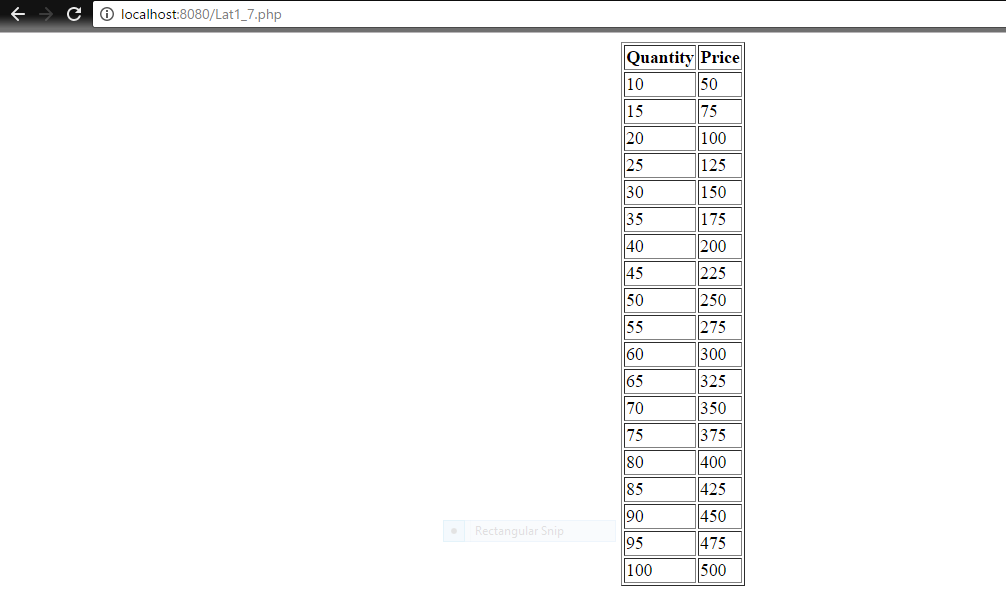
1. **Soal**

Tuliskan code dibawah ini, simpan dalam file Lat1\_7.php

|  |  |
| --- | --- |
|  | **<?php**  $brush\_price = 5;  echo "<table border=\"1\" align=\"center\">";  echo "<tr><th>Quantity</th>";  echo "<th>Price</th></tr>";  for ( $counter = 10; $counter <= 100; $counter += 10) {  echo "<tr><td>";  echo $counter;  echo "</td><td>";  echo $brush\_price \* $counter;  echo "</td></tr>";  }  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?

1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **<?php**  **$brush\_price = 5;**  **echo "<table border=\"1\" align=\"center\">";**  **echo "<tr><th>Quantity</th>";**  **echo "<th>Price</th></tr>";**  **for ( $counter = 10; $counter <= 100; $counter += 5) {**  **echo "<tr><td>";**  **echo $counter;**  **echo "</td><td>";**  **echo $brush\_price \* $counter;**  **echo "</td></tr>";**  **}**  **echo "</table>";**  **?>** |

1. **Penjelasan**

Apabila inkremen diganti dengan $counter+=5 maka perulangan terjadi sebanyak 18 kali untuk memenuhi nilai akhir 100 tadi. Alur kerja FOR melakukan perulangan dengan 3 parameter, pertama yaitu parameter awal, kedua adalah batas akhir parameter, dan yang ketiga adalah berapa banyak nilai yang dijumlahkan. Alur kerja WHILE melakukan perulangan dengan memberi satu persyaratan diawal. Jika persyaratannya terpenuhi maka akan terus terjadi perulangan sampai syarat tersebut tidak terpenuhi. Alur Kerja DO WHILE melakukan perulangan dengan menjalankan isi argumen terlebih dahulu. Setelah itu jika hasil dari argumen memenuhi while maka akan dilakukan perulangan sampai while tidak terpenuhi.

dijalankan sekali dulu baru dicek kondisi diakhir jika sesuai maka program akan kembali berulang.