

PHP

PROGATE



[Date]

[Company name]

[Company address]

Contents

[BAB 1 Apa itu PHP? 1](#_Toc87641353)

[1.1 Pendahuluan 2](#_Toc87641354)

[1.1.1 Apa Itu PHP? 2](#_Toc87641355)

[1.1.2 Bagaimana Menulis PHP Code 2](#_Toc87641356)

[1.1.3 Sintaksis PHP 2](#_Toc87641357)

[1.1.4 Output String 2](#_Toc87641358)

[1.1.5 Aritmatika 3](#_Toc87641359)

[1.1.6 Instruksi 3](#_Toc87641360)

[BAB 2 Dasar-Dasar PHP 4](#_Toc87641361)

[2.1 Tipe Data 5](#_Toc87641362)

[2.1.1 Tipe Data 5](#_Toc87641363)

[2.1.2 Membandingkan String dan Numerik 5](#_Toc87641364)

[2.2 Variable 5](#_Toc87641365)

[2.2.1 Apa Itu Variable? 5](#_Toc87641366)

[2.2.2 Peranan Variable 5](#_Toc87641367)

[2.2.3 Bagaimana Cara Memberi Nama Variable 6](#_Toc87641368)

[2.2.4 Instruksi 6](#_Toc87641369)

[2.3 Memperbarui Nilai Variable 6](#_Toc87641370)

[2.3.1 Memperbarui Variable 6](#_Toc87641371)

[2.3.2 Menambah Angka ke Variable 6](#_Toc87641372)

[2.3.3 Menambah Angka ke Variable (2) 7](#_Toc87641373)

[2.3.4 Menambah Angka ke Variable (3) 7](#_Toc87641374)

[2.3.5 Instruksi 7](#_Toc87641375)

[2.4 Menggabungkan String 8](#_Toc87641376)

[2.4.1 Penggabungan String 8](#_Toc87641377)

[2.4.2 Penggabungan String (2) 8](#_Toc87641378)

[2.4.3 Substitusi Variable 8](#_Toc87641379)

[BAB 3 Statement Kondisiona 9](#_Toc87641380)

[3.1 If 10](#_Toc87641381)

[3.1.1 Statement if 10](#_Toc87641382)

[3.1.2 Membandingkan Angka 10](#_Toc87641383)

[3.1.3 Statement if dan Nilai Boolean 10](#_Toc87641384)

[3.1.4 Instruksi 11](#_Toc87641385)

[3.2 Else dan Elseif 11](#_Toc87641386)

[3.2.1 else 11](#_Toc87641387)

[3.2.2 elseif 11](#_Toc87641388)

[3.2.3 Instruksi 11](#_Toc87641389)

[3.3 && dan || 12](#_Toc87641390)

[3.3.1 Menggabungkan Statement Kondisional 12](#_Toc87641391)

[3.3.2 Meniadakan Kondisi 12](#_Toc87641392)

[3.3.3 Instruksi 12](#_Toc87641393)

[3.4 Switch 13](#_Toc87641394)

[3.4.1 Statement switch 13](#_Toc87641395)

[3.4.2 Statement switch - break 13](#_Toc87641396)

[3.4.3 Instruksi 14](#_Toc87641397)

[BAB 4 Array 15](#_Toc87641398)

[4.1 Apa Itu Array? 16](#_Toc87641399)

[4.1.1 Membuat Array 16](#_Toc87641400)

[4.1.2 Menambah dan Menimpa Nilai 16](#_Toc87641401)

[4.1.3 Instruksi 16](#_Toc87641402)

[4.2 Array Associative 16](#_Toc87641403)

[4.2.1 Membuat Array Associative 16](#_Toc87641404)

[4.2.2 Bagaimana Menggunakan Array Associative 17](#_Toc87641405)

[4.2.3 Instruksi 17](#_Toc87641406)

[BAB 5 Loop 18](#_Toc87641407)

[5.1 For 19](#_Toc87641408)

[5.1.1 Loop dan Iterasi 19](#_Toc87641409)

[5.1.2 Loop for 19](#_Toc87641410)

[5.1.3 Instruksi 19](#_Toc87641411)

[5.2 while 20](#_Toc87641412)

[5.2.1 Loop while 20](#_Toc87641413)

[5.2.2 Loop Infinite (Tak Terbatas) 20](#_Toc87641414)

[5.2.3 Instruksi 20](#_Toc87641415)

[5.3 break 20](#_Toc87641416)

[5.3.1 break 20](#_Toc87641417)

[5.3.2 Instruksi 21](#_Toc87641418)

[5.4 continue 21](#_Toc87641419)

[5.4.1 continue 21](#_Toc87641420)

[5.4.2 Instruksi 21](#_Toc87641421)

[5.5 foreach 21](#_Toc87641422)

[5.5.1 foreach 21](#_Toc87641423)

[5.5.2 Bagaimana Menggunakan foreach 22](#_Toc87641424)

[5.5.3 Bagaimana Menggunakan foreach (2) 22](#_Toc87641425)

[5.5.4 Instruksi 22](#_Toc87641426)

[BAB 6 Menggunakan Function 23](#_Toc87641427)

[6.1 Function 24](#_Toc87641428)

[6.1.1 Menggunakan Function 24](#_Toc87641429)

[6.1.2 Menggunakan Function 24](#_Toc87641430)

[6.1.3 Instruksi 24](#_Toc87641431)

[6.2 Membuat Function 24](#_Toc87641432)

[6.2.1 Membuat Function 24](#_Toc87641433)

[6.2.2 Bagaimana Membuat Function 25](#_Toc87641434)

[6.2.3 Argument 25](#_Toc87641435)

[6.2.4 Instruksi 25](#_Toc87641436)

[6.3 Tipe-Tipe Return 26](#_Toc87641437)

[6.3.1 Nilai Return 26](#_Toc87641438)

[6.3.2 Instruksi 26](#_Toc87641439)

[BAB 7 Membuat Formulir Kontak 28](#_Toc87641440)

[7.1 <form> 29](#_Toc87641441)

[7.1.1 Membuat Formulir 29](#_Toc87641442)

[7.1.2 Menyiapkan Formulir 29](#_Toc87641443)

[7.1.3 Instruksi 29](#_Toc87641444)

[7.2 Kotak Teks 29](#_Toc87641445)

[7.2.1 Membuat Kotak Teks 29](#_Toc87641446)

[7.2.2 Membuat Kotak Teks (2) 29](#_Toc87641447)

[7.2.3 Instruksi 30](#_Toc87641448)

[7.3 $\_POST 30](#_Toc87641449)

[7.3.1 Membuat Tombol Kirim 30](#_Toc87641450)

[7.3.2 Menerima Data dari Formulir 30](#_Toc87641451)

[7.3.3 Instruksi 30](#_Toc87641452)

[7.4 <select> 31](#_Toc87641453)

[7.4.1 Bagaimana untuk Membuat Kotak Pilihan 31](#_Toc87641454)

[7.4.2 Mengirim Nilai yang Dipilih 31](#_Toc87641455)

[7.4.3 Instruksi 31](#_Toc87641456)

[7.5 Menyelesaikan Formulir 32](#_Toc87641457)

[7.5.1 Menyelesaikan Formulir 32](#_Toc87641458)

[7.5.2 Instruksi 32](#_Toc87641459)

# Apa itu PHP?

## Pendahuluan

Mari Mempelajari PHP

Mempelajari PHP akan memudahkan Anda untuk membuat fungsi yang dibutuhkan untuk membangun aplikasi web

seperti menangani, menyimpan, dan menampilkan input pengguna.

Kita akan membuat formulir kontak pada pelajaran ini seperti yang ditunjukkan di bawah.

### Apa Itu PHP?

Dengan HTML, ada limitasi bagaimana kita mengontrol apa yang ditampilkan. Dengan menggunakan PHP, Anda dapat mengubah teks yang ditampilkan tergantung audiens dan situasi.

### Bagaimana Menulis PHP Code

Code PHP dapat tertanam di dalam HTML. Dengan menggunakan tag PHP spesial <?php ... ?>, Anda dapat menulis instruksi didalamnya. Code yang ditulis di dalam <?php ... ?> akan diubah menjadi HTML dan ditampilkan.

Code PHP

<html>

<head></head>

<body>

<h1>PHP</h1>

<?php echo '<h2>Pemula</h2>'; ?>

</body>

</html>

### Sintaksis PHP

Titik koma ; digunakan untuk memisahkan statement di PHP. Akan ada error jika sintaksis Anda kurang titik koma, jadi mari berhati-hati! Selain itu, baris-baris yang dimulai dengan // disebut komentar dan tidak dijalankan. Komentar tidak akan mempengaruhi code Anda, namun sangat berguna untuk mendokumentasikan dan membuat catatan.

index.php

<?php

//Baris ini adalah komentar

echo 'Halo';

echo 'Belajar PHP!';

?>

### Output String

Anda dapat menggunakan echo untuk mencetak karakter yang disebut string. Gunakan tanda kutip satu ' atau kutip dua " untuk mencetak mereka.

Mencetak String

<?php echo 'Hello, World!'; ?>

//cetak teks didalam tanda kutip

Tutup String dengan tanda kutip satu (') atau dua (")

<?php echo 'Hello, World!'; ?>

<?php echo "Hello, PHP!"; ?>

### Aritmatika

Kalkulasi adalah hal terbaik yang bisa dilakukan oleh komputer! Dalam pemrograman, Anda juga dapat melakukan kalkulasi dengan simbol-simbol yang dikenal sebagai operator aritmatika. Simbol-simbol yang digunakan untuk perhitungan dalam PHP adalah sebagai berikut: + (tambah), - (kurang), \* (kali), dan / (bagi). Anda juga dapat menghitung sisa pembagian dengan simbol %.

index.php

echo 7 + 3; //Hasil: 10

echo 10 - 4; //Hasil: 6

echo 6 \* 3; //Hasil: 18

echo 9 / 3; //Hasil: 3

echo 5 % 2; //Hasil: 1

### Instruksi

Mari mulai dengan menulis beberapa code PHP!

Ikuti instruksi di bawah sembari menulis code Anda.

Setelah menulis code, lihat hasilnya dengan mengecek Jawaban Anda di sudut kanan atas.

index.php

Gunakan echo untuk print

Hello, world!

Lihat Slide

index.php

Gunakan echo untuk print

7 \* 2

index.php

Gunakan echo untuk print

8 % 3

# Dasar-Dasar PHP

## Tipe Data

### Tipe Data

Ada tipe-tipe data berbeda yang digunakan dalam PHP seperti angka dan karakter. Data teks seperti 'Anak' atau 'A' disebut string dan angka seperti 42 atau 3.1415 disebut tipe data numerik.

### Membandingkan String dan Numerik

5 + 2 menjadi 7 saat dicetak sebagai nilai numeric. Namun, jika Anda menggunakan tanda petik satu atau dua, nilai akan diinterpretasikan sebagai string '5 + 2'.

index.php

<?php

echo 5 + 2; //kalkulasi numerik

echo '5 + 2'; //string

?>

Instruksi

index.php

Gunakan echo untuk mencetak nilai numerik hasil dari kalkulasi

5 + 7

index.php

Gunakan echo untuk mencetak

5 + 7

sebagai string.

## Variable

### Apa Itu Variable?

Sekarang mari kita belajar konsep penting dalam pemrograman yang disebut variable. Variable diibaratkan kotak untuk menyimpan data. Kita mendefinisikan variable dengan menambah simbol $ diawal. Semua tipe data dapat disimpan menggunakan sintaksis $variableName = nilai;. Dalam pemrograman, = artinya untuk menentukan nilai di kanan pada apa yang terletak di sebelah kiri.

index.php

<?php

$name = 'Bob';

echo $name; //tetapkan 'Bob' ke variable $name

?>

### Peranan Variable

Ada 2 keuntungan menggunakan variable: (1) Anda dapat memberi nama pada data untuk mendeskripsikannya; dan (2) Anda dapat memperbarui nilai variable dari satu tempat dan dapat digunakan berkali-kali pada code Anda.

Code Dengan Variable

$length = 7;

$area = $length \* $length; //data yang sama dapat digunakan berulang kali

echo $area;

Code tanpa Varaiable

$area = 7 \* 7; //harus mengubah nilai berkali-kali jika terjadi perubahan

echo $area;

### Bagaimana Cara Memberi Nama Variable

Ada beberapa peraturan dan perjanjian yang harus Anda ikuti saat mendeklarasikan variable. Anda tidak dapat menggunakan angka pada awalan nama variable. Selain itu, penting untuk memilih nama variable yang dengan jelas mendeskripsikan jenis data yang tersimpan dalam variable tersebut. Kita harus konsisten dalam menulis nama variable.

Nama Variable yang Baik

$date .. huruf kecil

$user\_name .. pisahkan kata dengan underscore

$userName .. pisahkan kata dengan huruf kapital

Nama Variable yang Buruk

$a .. tidak deskriptif

$1name .. dimulai dengan angka

### Instruksi

index.php

Deklarasikan variable

$fruit

dan tentukan

Apel

ke variable tersebut.

index.php

Gunakan echo untuk print variable $fruit.

index.php

Deklarasikan variable

$sum

dan tentukan

8 + 9

ke variable tersebut.

index.php

Gunakan echo untuk print variable $sum.

## Memperbarui Nilai Variable

### Memperbarui Variable

Sebuah variable juga dapat diperbarui untuk mengubah nilai yang dikandungnya. Seperti saat Anda menentukan nilai lain pada variable, nilai baru ini menimpa konten asalnya.

index.php

$num = 3;

echo $num;

$num = 5; //perbarui nilai dari variable num

echo $num;

### Menambah Angka ke Variable

Apa yang seharusnya dilakukan, contohnya, saat kita ingin menambah 10 ke suatu variable? Kita hanya perlu untuk menambah angka ke variable dan menentukan hasilnya kembali ke variable tersebut. Kita dapat melakukan kalkulasi lain seperti pengurangan dengan cara yang sama. Dalam PHP, = artinya menentukan, bukan indikasi bahwa nilai kanan dan kiri sama.

index.php

$x = 1;

$x = $x + 10; //tambahkan ke variable $x dan tetapkan nilainya ke variable tersebut

echo $x;

### Menambah Angka ke Variable (2)

Faktanya, ada cara lebih singkat dalam menulis saat menambahkan angka ke variable. Seperti ditunjukkan di bawah, Anda dapat menggunakan singkatan untuk semua kalkulasi.

Standar Singkatan

$x = $x + 10 $x += 10

$x = $x - 10 $x -= 10

$x = $x \* 10 $x \*= 10

$x = $x / 10 $x /= 10

$x = $x % 10 $x %= 10

### Menambah Angka ke Variable (3)

Saat Anda menambah atau mengurangi 1, Anda dapat mempersingkatnya dengan operator penambahan ++ dan operator pengurangan --. Saat Anda meletakkan operator penambahan sebelum variable, kalkulasi selesai sebelum echo. Akan tetapi, jika operator penambahan diletakkan setelah variable, kalkulasi dijalankan setelah echo.

Singkatan tambahan untuk penambahan & pengurangan 1

index.php

$x += 1; -> $x++;

$x -= 1; -> $x--;

index.php

$x = 3;

$y = 3;

echo ++$x; //tambah 1 sebelum menjalankan echo

echo $y++; //tambah 1 setelah menjalankan echo

### Instruksi

Beberapa variable sudah dideklarasikan untuk Anda.

index.php

・Perbarui $x dengan menambah 10.

・Gunakan echo untuk mencetak $x.

index.php

・Perbarui $y dengan mengalikan 5.

・Gunakan echo untuk mencetak $y.

index.php

・Perbarui $a dengan menambah 1.

・Gunakan echo untuk mencetak $a.

index.php

・Perbarui $b dengan mengurangi 1.

・Gunakan echo untuk mencetak $b.

## Menggabungkan String

### Penggabungan String

Anda dapat menggabungkan string menggunakan titik(.). Ini disebut operator dot. Ada 3 jenis penggabungan string:

(1) string dengan string, (2) variable string dengan variable string, dan (3) string dengan variable string.

index.php

$lesson = 'PHP ';

echo $lesson.'Pemula'; //pisahkan dengan titik

$level = 'Pemula';

echo $lesson.$level;

### Penggabungan String (2)

Anda dapat menghilangkan penggabungan variable dan string menggunakan .=.

index.php

$name = 'Ninja ';

$name .= 'Ken'; //sama seperti $name = $name."Ken"

echo $name;

### Substitusi Variable

Di dalam string di dalam tanda kutip dua ", Anda dapat menyisipkan sebuah variable dengan meletakkan nama variable di dalam tanda kurung { }. Ini disebut substitusi variable. Jika Anda menggunakan petik satu ', variable tidak akan disubstitusikan karena akan diinterpretasikan sebagai string bahkan jika itu di dalam tanda kurung.

index.php

$name = 'Ninja Ken';

echo "Halo, {$name}" //diganti dengan nilai dari $name

echo 'Halo, {$name}' //{$name} dicetak sebagai string

Variable $name sudah dideklarasikan untuk Anda dalam file index.php.

index.php

Gunakan echo untuk mencetak

Halo, \_\_\_\_

menggunakan variable $name.

(Ganti \_\_\_\_ dengan variable $name.)

# Statement Kondisiona

## If

### Statement if

Statement if memungkinkan kita untuk membuat percabangan kondisional. Jika Anda meletakkan kondisi di dalam () setelah if, code diantara tanda kurung { } akan dijalankan hanya jika kondisi benar.

index.php

$x = 20;

if ($x > 10) {

//jika kondisi benar

echo '$x lebih besar dari 10';

//code ini dijalankan

index.php

$x = 20;

if ($x > 30) {

//jika kondisi ini salah, code ini TIDAK akan dijalankan

echo '$x lebih besar dari 10';

### Membandingkan Angka

Simbol-simbol yang digunakan untuk membandingkan dua nilai, seperti > dari slide sebelumnya, disebut operator perbandingan. Nilai dari kondisi operator adalah true atau false. Operator yang umum ditunjukkan di bawah.

x == y

true jika nilai x dan y sama.

false jika nilai x dan y tidak sama.

x != y

true jika nilai x dan y tidak sama.

false jika nilai x dan y sama.

x < y true jika nilai x lebih kecil dari y.

x <= y true jika nilai x lebih kecil dari atau sama dengan y.

x > y true jika nilai x lebih besar dari y.

x >= y true jika nilai x lebih besar dari atau sama dengan y.

### Statement if dan Nilai Boolean

Kondisi dari operator perbandingan akan digantikan dengan sebuah nilai boolean. Boolean adalah sebuah tipe data yang hanya memiliki dua nilai: true dan false. Statement if hanya akan menjalankan code diantara tanda { } ketika kondisinya true.

index.php

if (true) {

//selalu menjalankan code didalam if

echo 'Kondisi adalah true';

}

index.php

$x = 20;

if ($x > 10) {

//jika kondisi benar

echo '$x lebih besar dari 10';

//code ini dijalankan

}

if ($x > 30) {

//jika kondisi ini salah, code ini TIDAK akan dijalankan

echo '$x lebih besar dari 10';

### Instruksi

Variable $x dan $y sudah dideklarasikan di dalam file index.php dengan nilai 99 \* 99 dan 77 \* 77.

index.php

Menggunakan statement if, jika variable $x lebih besar dari 9800, gunakan echo untuk mencetak

Variable $x lebih besar dari 9800.

index.php

Menggunakan statement if, jika variable $y lebih besar dari 6000, gunakan echo untuk mencetak

Variable $y lebih besar dari 6000.

## Else dan Elseif

### else

Jika Anda menggunakan else dalam kombinasi statement if, Anda dapat menambahkan code untuk ketika kondisi adalah false. Jika statement if yang cocok salah, code di else akan dijalankan.

index.php

$x = 20;

if ($x == 30) {

echo '$x adalah 30'; //code ini tidak dijalankan

} else {

echo '$x bukan 30'; //code ini dijalankan

### elseif

Anda dapat menggunakan statement elseif untuk menambahkan kondisi alternatif ke control flow. Contoh kondisi alternatif mungkin: A akan dijalankan jika kondisi X adalah true; elseif Y adalah true mengingat X adalah false, B akan dieksekusi; jika semua kondisi false, C akan dijalankan.

$x = 20;

if ($x < 30) {

echo '$x adalah 30'; //code ini dijalankan karena kondisi benar

} elseif ($x >= 20) {

echo '$x lebih besar dari 20 atau sama dengan 20'; //walaupun kondisi benar, ini tidak dijalankan

} else {

echo '$x lebih kecil dari 20';

}

### Instruksi

index.php

Deklarasikan variable

$age

dan tentukan umur Anda ke variable tersebut.

index.php

Menggunakan statement if-else, jika variable $age lebih besar atau sama dengan 30, gunakan echo untuk mencetak

Kamu berumur 30 tahun atau lebih.

Jika variable $age kurang dari 30, gunakan echo untuk mencetak

Kamu berumur dibawah 30 tahun.

## && dan ||

### Menggabungkan Statement Kondisional

Dan && dan atau || dikenal sebagai operator logika. Mereka digunakan untuk menggabungkan beberapa kondisi. Hasil dari statement dengan && menjadi true hanya jika ekspresi kiri dan kanan adalah true. Hasil statement dengan || menjadi true ketika salah satu atau kedua ekspresi kiri dan kanan adalah true.

index.php

$x = 20;

if ($x > 10 && $x < 30) {

echo '$x lebih besar dari 10 dan lebih kecil dari 30';

}

if ($x < 10 || $x > 30) {

echo '$x lebih kecil dari 10 atau lebih besar dari 30';

}

### Meniadakan Kondisi

Tanda seru ! juga merupakan operator logika. Anda dapat meniadakan sebuah kondisi dengan menggunakan !, dimana hasilnya menjadi false jika ekspresinya true, dan true jika ekspresinya false.

index.php

$x = 20;

if (!($x == 30)) {

echo '$x tidak sama dengan 30'; //tidak sma dengan (membalikkan logika kondisi)

}

### Instruksi

Variable $x sudah disiapkan untuk Anda di dalam file index.php dengan nilai 1071.

index.php

Dengan menggunakan statement if-elseif-else, buat control flow menggunakan logika berikut:

if

・variable $x adalah kelipatan 3 dan kelipatan 7, gunakan echo untuk mencetak

x adalah kelipatan 3 dan kelipatan 7.

Else if

・variable $x adalah kelipatan 3, gunakan echo untuk mencetak

x adalah kelipatan 3 tetapi bukan kelipatan 7.

・variable $x adalah kelipatan 7, gunakan echo untuk mencetak

x adalah kelipatan 7 tetapi bukan kelipatan 3.

Else

・Gunakan echo untuk mencetak

x bukan kelipatan 7 atau kelipatan 3.

Catatan: Jika variable $x adalah kelipatan 3, sisa dari $x dibagi 3 adalah 0, yang artinya adalah $x % 3 == 0.

## Switch

### Statement switch

Statement switch adalah cara lain untuk mengontrol flow code Anda agar lebih mudah dibaca dan dipahami, terutama saat percabangan kondisional menjadi rumit. Statement switch biasanya memiliki banyak case. Case pertama yang cocok dengan nilai kondisi akan dijalankan. Jika tidak ada yang cocok, case default dijalankan.

Control Flow Menggunakan Statement if

if ($coin ==0) {

echo 'Kepala';

} elseif ($coin == 1) {

echo 'Ekor';

} else {

echo 'Error';

}

Control Flow Menggunakan Statement switch

switch ($coin) {

case 0:

echo 'Kepala';

break;

case 1:

echo 'Ekor';

break;

default:

echo 'Error';

break;

}

### Statement switch - break

Dengan statement switch, setiap case membutuhkan statement break di akhir. Statement break akan menghentikan statement switch yang sedang berjalan. Tanpanya, statement switch akan berlanjut ke case berikutnya hingga statement break ditemukan. Jadi, pastikan untuk meletakkan satu di akhir setiap case!

index.php

switch ($coin) {

case 0:

echo 'Kepala'; //break tidak ditemukan: case berikutnya dijalankan walaupun kondisinya tidak cocok

case 1:

echo 'Ekor';

break;

default:

echo 'Error';

break;

}

index.php

switch ($coin) {

case 0:

echo 'Kepala';

break; //break: program menghentikan statement switch

case 1:

echo 'Ekor';

break;

default:

echo 'Error';

break;

}

### Instruksi

Mari membuat permainan meramal menggunakan statement switch.

index.php

Tentukan nomor ke variable $num. $num seharusnya lebih besar atau sama dengan 0.

index.php

Tentukan sisa dari $num dibagi 3 menggunakan % ke variable $remainder.

index.php

Dengan menggunakan statement switch:

・Jika $remainder adalah 0, gunakan echo untuk mencetak

Sangat beruntung!!

・Jika $remainder adalah 1, gunakan echo untuk mencetak

Beruntung!

・Jika $remainder adalah 2, gunakan echo untuk mencetak

Sedikit beruntung.

・Dalam case lain, gunakan echo untuk mencetak

Tidak beruntung...

# Array

## Apa Itu Array?

Anda dapat menyimpan beberapa nilai secara bersamaan menggunakan array, sedangkan variable hanya dapat menangani satu nilai seperti yang kita pelajari sebelumnya. Sebuah array dapat dibayangkan seperti kotak dengan partisi didalamnya; setiap ruang berisi data, dan nama untuk setiap ruang diberi nomor index (0, 1, 2…).

### Membuat Array

Sintaksis dasar untuk mendeklarasikan sebuah array adalah sebagai berikut: $arrayName = array(nilai1, nilai2,…);. Nomor index ditetapkan untuk setiap element array dalam urutan dari 0 hingga element terakhir. Untuk mengambil data dari sebuah array, kita dapat menggunakan nomor index seperti: $arrayName[nomorIndex].

Membuat sebuah Array

$names = array('John', 'Kate', 'Bob');

//index 0 1 2

Mengakses elemnt didalam array

echo $names[0]; //hasil: John

echo $names[1]; //hasil: Kate

### Menambah dan Menimpa Nilai

Untuk menambahkan element ke akhir array, tulis seperti berikut ini: $arrayName[] = nilai;. Anda juga bisa menimpa nilai dalam array dengan menentukan nomor index yang sama.

index.php

$names = array('John', 'Kate', 'Bob');

$names[] = 'Mary'; //tambahkan element ke belakang array

echo $names[3];

$names[1] = 'Jane'; //mengubah element di index 1

echo $names[1];

### Instruksi

index.php

Definisikan array yang bernama $colors dengan element berikut:

Merah

Biru

Kuning

index.php

Gunakan echo untuk print element pada index 0.

index.php

Tambahkan

Putih

pada akhir $colors dan gunakan echo untuk print element terakhir.

## Array Associative

### Membuat Array Associative

Array associative memungkinkan kita untuk mengelola kumpulan data seperti array biasa. Perbedaannya adalah daripada menggunakan nomor index untuk mengelola element individu, Anda dapat menentukan nilai sebagai kunci. Kunci bisa berbentuk string. Anda dapat memasangkan kunci dan nilainya dengan => sebagai berikut: arrayName = array('namaKunci' => 'nilai1', ...);.

Array Associative

$user = array(

'name' => 'Ken',

'age' => 14,

'gender'=> 'pria'

//kunci //nilai

);

### Bagaimana Menggunakan Array Associative

Untuk mengambil nilai dari sebuah array associative, kita menggunakan "kunci" yang sesuai sebagai berikut: arrayName[kunci]. Anda juga dapat menambahkan data ke array associative, seperti yang ditunjukkan pada gambar di bawah ini, dengan menggunakan: arrayName[kunci] = nilai;.

Mengakses Element di Array Associative

$user = array(

'name' => 'Ken',

'age' => 14,

);

echo $use['name'];

Menambahkan Elemtn ke Array Associative

$use['level'] = 'pemula';

//kunci //nilai

### Instruksi

index.php

Definisikan array associative yang bernama

$scores

dengan element berikut:

・kunci

Matematika

, nilai

70

・kunci

Bahasa

, nilai

90

・kunci

Sains

, nilai

80

# Loop

## For

### Loop dan Iterasi

Kami menggunakan loop untuk melakukan sesuatu berulang kali. Misalnya, Anda ingin mencetak angka dari 1 hingga 100. Dengan menggunakan loop, Anda dapat mengurangi code panjang dan berulang menjadi hanya beberapa baris.

Mencetak nomor dari 1 hingga 100 dengan echo

echo 1;

echo 2;

echo 3;

.

.

.

echo 100;

Menggunakan loop for

for ($ = 1; $i <= 100, $i++) {

echo $i;

}

### Loop for

Anda dapat mengiterasikan (mengulang) code menggunakan loop for. Variable $i diberikan nilai 1 sebagai nilai awal, kemudian echo $i; dijalankan. Setelah itu, nilai $i diperbarui ($i++) dan echo $i; dijalankan berulang kali hingga kondisi loop tidak lagi terpenuhi.

Contoh Loop for

for ($i = 1; $i <= 100; $i++) {

echo $i;

}

Cara kerja loop

1. inisiasi variable

2. memeriksa kondisi

3. menjalankan code didalam loop

4. memperbarui variable

### Instruksi

index.php

Gunakan loop for dan echo untuk mencetak nomor dari 51 sampai 100.

index.php

Untuk mencetak tiap nomor pada baris baru, kita dapat menggunakan <br> sebagai berikut:

・Gabungkan string

echo $i.'<br>';

atau

・Gunakan echo

echo $i;

echo '<br>';

## while

### Loop while

Loop while memungkinkan Anda untuk mengulang code seperti loop for. Ketika sebuah kondisi ditentukan, code dalam loop akan dijalankan berulang kali selama nilainya adalah true. Berbeda dari loop for, code untuk yang menaikkan variable, $i++;, harus ditulis dalam loop. Ini adalah perbedaan penting antara loop while dan for.

Contoh Loop while

$i = 1; //1. inisialisasi variable

while ($i <= 100) { //2. memeriksa kondisi

echo $i; //3. mengulang code

$i++; //4. memperbarui variable

}

### Loop Infinite (Tak Terbatas)

Jika Anda lupa memperbarui nilai variable yang digunakan untuk kondisi di akhir loop while, ini akan menyebabkan loop infinite karena kondisinya akan selalu true. Loop infinite akan memberikan beban besar pada komputer, jadi pastikan ada kondisi false di titik tertentu sebelum menjalankannya.

Loop Infinite

$i = 1;

while ($i <= 1000 {

//loop akan diulang selamanya karena kondisi akan selalu true

echo $i;

}

### Instruksi

Mari deklarasikan variable untuk menangani loop while.

index.php

Deklarasikan variable $i dan tentukan

2

ke variable tersebut.

index.php

Gunakan loop while, variable $i, dan echo untuk mencetak semua bilangan genap diantara 2 dan 100.

index.php

Untuk mencetak tiap nomor pada baris baru, kita dapat menggunakan <br> sebagai berikut:

・Gabungkan string

echo $i.'<br>';

atau

・Gunakan echo

echo $i;

echo '<br>';

## break

### break

Statement break secara paksa akan mengakhiri loop dan digunakan dalam statement berulang seperti loop (for, while, foreach, dll). Statement break umumnya digunakan dalam kombinasi dengan statement conditional seperti statement if.

break - memaksa loop untuk berhenti

for ($i = 1; $i <= 10; $i++) {

if ($i > 5) {

break; //keluar dari loop ketika $i sama dengan 6

}

echo $i;

}

### Instruksi

index.php

Gunakan loop for dan echo untuk mencetak nomor dari 1 sampai 1000.

Akan tetapi, hentikan loop menggunakan break jika nomor lebih besar dari 500.

Pastikan untuk mencetak tiap nomor pada baris baru!

## continue

### continue

Ketika statement break mengeluarkan kita dari loop, statement continue hanya melewatkan iterasi saat ini tetapi tetap melanjutkan loop. Statement continue juga bisa digunakan dalam statement iteratif seperti for, while,foreach, dll.

continue - melompati loop saat ini

for ($i = 1; $i <= 10; $i++) {

if ($i % 3 == 0) {

continue; //ketika $i adalah kelipatan 3, loop saat ini akan diloncati

}

echo $i;

}

### Instruksi

index.php

Gunakan loop for dan echo untuk mencetak nomor 1 sampai 1000. Akan tetapi, lewatkan nomor kelipatan 3 menggunakan continue.

Jangan lupa untuk mencetak tiap nomor pada baris baru!

## foreach

### foreach

Loop foreach melakukan iterasi pada array dengan index atau array associative secara berurutan dari nilai pertama. Anda dapat mengambil nilai dalam array satu per satu seperti yang ditunjukkan di bawah ini. Nilai ditentukan secara berurutan ke variable setelah kata kunci as pada awal setiap loop. Nama variable setelah as dapat berupa apa saja yang Anda inginkan.

index.php

$town = array('Tokyo', 'London', 'New York');

foreach ($town as $town) {

echo $town.'';

}

### Bagaimana Menggunakan foreach

Dalam loop foreach, nilai array secara berurutan ditetapkan ke variable kunci ($key) dan variable nilai ($value), lalu code dalam loop akan berulang kali dijalankan. Nomor index (untuk array dengan index) atau kunci (untuk array associative) diberikan ke variable kunci. Namun, perhatikan bahwa variable kunci bersifat opsional.

index.php

foreach ($array as $value) {

//menjalankan code..

}

foreach ($array as $key => $value) { //$key dapat dihapus

//menjalankan code..

}

### Bagaimana Menggunakan foreach (2)

Mari kita lihat contoh loop foreach. Di bawah ini adalah contoh dengan array associative yang memiliki nama buah sebagai kunci dan warna sebagai nilai. Dalam iterasi pertama loop, "Apel" dan "Merah" digunakan sebagai $key dan$value. Pada iterasi berikutnya, "Pisang" dan "Kuning" digunakan.

index.php

$colors = array(

'apel' => 'merah',

'pisang => 'kuning',

'anggur' => 'ungu',

);

foreach ($colors as $key => $value) {

echo $key.':'.$value.'';

} //Hasil: apel:merah pisang:kuning anggur:ungu

### Instruksi

Array associative yang bernama $scores sudah dideklarasikan untuk Anda di dalam file index.php.

index.php

Tulis loop foreach menggunakan array associative $scores dimana variable $key sebagai kunci dan variable $value sebagai nilai.

index.php

Cetak skor tiap subjek seperti Matematika: 70 .

# Menggunakan Function

## Function

### Menggunakan Function

Function adalah bagian code yang dirancang untuk menyelesaikan tugas tertentu dan mereturn hasil. Beberapa function umum dan berguna sudah tertanam di PHP; function ini disebut built-in function. Misalnya strlen mengembalikan jumlah karakter dalam sebuah string. Di sini, nilai dalam tanda kurung () disebut argument.

index.php

echo strlen('Progate');

//function yang mereturn panjang sebuah string

//Hasil: 7

index.php

$language = 'PHP';

echo strlen($language);

//dapat meneruskan variable sebagai variable

//Hasil: 3

### Menggunakan Function

Mari kita lihat beberapa built-in function (function bawaan) lainnya: count mengembalikan jumlah element dalam array; rand mengembalikan angka acak antara argument pertama dan argument kedua. Ada banyak function bawaan PHP lainnya, tetapi Anda tidak perlu menghafalnya. Lebih baik mencari dan mempelajari tentang function yang berguna untuk tujuan tertentu.

count() - menurun jumlah element di array

$town = array('Tokyo', 'London', 'New York');

echo count($towns);

//return 3

Hasil: 3

rand() - mengeluarkan angka acak

echo rand(1, 4);

//return angka acak antara 1 hingga 4

### Instruksi

Mari coba menggunakan function!

String $str dan array $array sudah didefinisikan untuk Anda di dalam file index.php.

index.php

Gunakan function strlen dan echo untuk print panjangnya $str.

index.php

Gunakan function count dan echo untuk mencetak jumlah element dari $array.

index.php

Gunakan function rand dan echo untuk mencetak angka acak diantara 10 dan 15.

## Membuat Function

### Membuat Function

Anda juga dapat membuat function sendiri. Saat Anda mengelompokkan code di satu tempat, Anda dapat membuat perubahan ke semua tempat yang digunakan sekaligus dengan mengedit function yang memuatnya. Ini jauh lebih mudah dikelola daripada mengubah banyak baris code yang terpisah dan hampir identik. Mari pelajari lebih lanjut tentang cara mendefinisikan function di slide berikutnya.

Menghitung area sebuah lingkaran

$radius1 = 3;

echo $radius1 \* $radius1 \* 3.14;

$radius2 = 5;

echo $radius2 \* $radius2 \* 3.14;

index.php

function printCircleArea($radius1) {

//mendefinisikan function

echo $radius1 \* $radius1 \* 3.14

}

//memanggil function

printCirlceArea(3); //hasil: 28.26

printCirlceArea(5); //hasil: 78.5

### Bagaimana Membuat Function

Gunakan sintaksis: function functionName() {...} untuk membuat function. Beri saja nama function Anda dan panggil menggunakan tanda kurung () seperti begitu: functionName().

Mendefinisikan function

function hello() {

echo 'Hello, world!';

}

Memanggil function

hello();//()dibutuhkan ketika memanggil function

//Hasil: Hello, world!

### Argument

Saat memanggil suatu function, Anda dapat mengirimkan nilai untuk digunakan di dalam function tersebut. Nilai yang diteruskan ke suatu function dikenal sebagai argument. Untuk menggunakan argument dalam suatu function, Anda perlu menentukan parameter yang sesuai untuk menerima argument dalam definisi function. Jika ada beberapa parameter, Anda perlu memisahkan masing-masing dengan koma ,.

Mendefinisikan function

function printSum($num1, $num2) {

echo $num1 + $num2;

}

Memanggil function

printSum(1,3);

function printSum($num1, $num2) {

echo $num1 + $num2;

}

### Instruksi

index.php

Definisikan function

hello

.

Di dalam function hello, gunakan echo untuk print

Hello, world!

.

index.php

Panggil function hello.

Lihat Slide

index.php

Definisikan function

printRectangleArea

dengan spesifikasi berikut:

・Parameter pertama adalah

$height

.

・Parameter kedua adalah

$width

.

Di dalam printRectangleArea, gunakan echo untuk print

$height \* $width

.

index.php

Panggil function printRectangleArea dengan argument

5

dan

10

.

## Tipe-Tipe Return

### Nilai Return

Sebuah function dapat mereturn nilai, dan nilai ini dikenal sebagai nilai return. Bagaikan menjalankan function dan bagian yang menjalankan function tersebut diganti dengan nilai yang di return. Nilai return ditentukan oleh return.

Mendifinisikan function

function getSum($num1, $num2) {

return $num1 + $num2;

}

Memanggil function

$sum = getSum(1, 3);

echo $sum;

function getsum($num1, $num2) {

return $num1 + $num2;

}

### Instruksi

index.php

Definisikan function

getCircleArea

dengan spesifikasi-spesifikasi berikut:

・Parameternya adalah

$radius

.

・Nilai return adalah

$radius \* $radius \* 3

. (Untuk menyederhanakan perhitungan, mari gunakan 3 sebagai nilai pi)

index.php

Panggil function getCircleArea dan lewatkan

5

sebagai argument. Lalu tentukan nilai return ke variable

$circleArea

.

index.php

Gunakan echo untuk print $circleArea.

# Membuat Formulir Kontak

## <form>

### Membuat Formulir

Untuk menyelesaikan, mari buat formulir kontak! Kita bisa membuat desain formulir dengan HTML dan kemudian menggunakan PHP untuk menangani nilai input.

### Menyiapkan Formulir

Saat membuat formulir, mari menggunakan tag HTML <form>. Untuk attribute action, tentukan URL untuk meneruskan data. Attribute method diberikan get atau post untuk mengirim nilai formulir. Saat menggunakan get, nilai yang akan dikirim ditampilkan di URL sedangkan saat menggunakanpost, nilainnya akan disembunyikan dari URL.

Mempersiapkan formulir

<form action="url" method="post">

//tulis code untuk formulir

</form>

### Instruksi

Kali ini, kita akan mengirim nilai form ke halaman yang bernama sent.php.

CSS sudah disiapkan untuk Anda, jadi mari kita tambahkan HTML yang sesuai.

index.php

Tambah tag <form>:

・ Atur attribute action ke

sent.php

・ Atur attribute method ke

post

## Kotak Teks

### Membuat Kotak Teks

Untuk membuat kotak teks, kita dapat menggunakan tag HTML <input type="text"> . Lalu, gunakan attribute name untuk mengambil nilai input. Attribute ini digunakan untuk menerima data formulir. Perhatikan bahwa tag <input> tidak membutuhkan tag penutup.

index.php

<form action="sent.php" method="post">

Masukkan email anda

<input type="text" name="email'>

//beri label pada data yang di input

</form>

### Membuat Kotak Teks (2)

Anda dapat membuat kotak teks untuk kalimat multi-baris dengan menggunakan tag HTML <textarea>. Mirip dengan tag <input>, attribute name bisa digunakan untuk mengambil nilai dari tag<textarea>. Perhatikan bahwa tidak seperti <input>, tag <textarea> memerlukan tag penutup.

index.php

<form action="sent.php" method="post">

Pesan

<textarea name="content"></textarea>

//beri label ke input

</form>

### Instruksi

index.php

Tambah tag <input>:

・Atur attribute type ke

text

・Atur attribute name ke

name

index.php

Tambah tag <textarea>:

・Atur attribute name ke

body

## $\_POST

### Membuat Tombol Kirim

Untuk membuat tombol kirim, Anda dapat menggunakan <input type="submit">. Nilai yang ditetapkan untuk attribute value akan ditampilkan pada tombol.

index.php

<form action="sent.php" method="post">

Masukkan email Anda disini

<input type="text" name="email" value="">

<input type="submit" value="Kirim">

</form>

### Menerima Data dari Formulir

Kita dapat menggunakan $\_POST untuk menerima nilai yang kita kirimkan di formulir. $\_POST adalah array associative. Oleh karena itu, kita bisa menerima nilai yang kita kirimkan dengan meletakkan attribute name dari <input> dan <textarea> dalam tanda kurung [].

sent.php (URL untuk mengirim data formulir)

echo $\_POST['name']; //hasil: Ninja Ken

echo $\_POST['email'];//hasil: ninjaken@progate.com

$\_POST adalah array associative

array(

'name' => 'Ninja Ken',

'email'=> 'ninjaken@prog-8.com',

)

### Instruksi

index.php

Tambah tag <input>:

・Atur attribute type ke

submit

・Atur attribute value ke

Kirim

Konten yang dikirim menggunakan method POST akan diterima oleh halaman sent.php.

sent.php

Di bawah dimana Nama ditampilkan, gunakan echo untuk mencetak nilai dari name yang dikirim menggunakan method POST.

Petunjuk

Jangan lupa untuk mengelilingi code dengan tag PHP.

sent.php

Di bawah Pesan, gunakan echo untuk mencetak nilai body menggunakan method POST.

Setelah menekan tombol refresh pada window Jawaban Anda, kirim formulir untuk mengonfirmasi bahwa nilai-nilai form sudah disimpan pada $\_POST.

## <select>

### Bagaimana untuk Membuat Kotak Pilihan

Kita dapat membuat kotak dropdown seperti yang ditunjukkan pada contoh di bawah ini dengan mengatur tag <option> di dalam tag <select>. Konten tag <option> ditampilkan sebagai pilihan.

index.php

<select>

<option>Apel</option>

<option>Pisang</option>

<option>Jeruk</option>

</select>

### Mengirim Nilai yang Dipilih

Mari kita lihat cara memasukkan nilai dari daftar dropdown. Tag <select> memiliki attribute name untuk menerima nilai dengan$\_POST. Sedangkan, attribute value dari tag <option> adalah nilai yang akan dikirim.

index.php

<form action="sent.php" method="post">

<select name="fruit"> //name adalah nama data yang digunakan untuk menerima nilai

<option value="Apel">Apel</option> //value adalah nilai yang dikirim

<option value="Pisang">Pisang</option>

<option value="Jeruk">Jeruk</option>

</select>

</form>

### Instruksi

index.php

Di bawah dimana Umur ditampilkan, tambahkan tag <select> dan atur attribute name ke

age

.

index.php

Pada tag <select> dengan name yang bernilai age, tambahkan 3 tag <option>:

・Pilihan 1:

Pilih

Atur attribute value

ke

pilih

・Pilihan 2:

20-29

Atur attribute value ke

20-29

・Pilihan 3:

30-39

Atur attribute value ke

30-39

Selanjutnya, mari menerima dan menampilkan pesan yang dikirim dengan method POST pada halaman sent.php.

sent.php

Di bawah dimana Umur ditampilkan, gunakan echo untuk mencetak nilai age yang dikirim menggunakan POST.

## Menyelesaikan Formulir

### Menyelesaikan Formulir

Ini latihan terakhir! Mari buat banyak tag <option> menggunakan loop dan variable substitusi. Saat menggunakan variable substitusi, pastikan untuk meletakkan code dalam tanda kutip ganda " ketika dibutuhkan.

index.php

for ($i = 1; $i < 4; $i++) {

echo "<option value='{$i}'>{$i}</option>";

}

Bekerja sama seperti yang di bawah

echo "<option value='1'>1</option>"

echo "<option value='2'>2</option>"

echo "<option value='3'>3</option>"

### Instruksi

Mari membuat pilihan untuk Umur secara lebih tepat dengan menambahkan pilihan untuk umur dari 6 sampai 100. Untuk melakukan ini, kita perlu untuk menulis tag <option> terpisah menggunakan loop.

index.php

Gunakan iterasi loop for dan echo untuk print tag <option> untuk setiap nomor dari 6 sampai 100.

index.php

Di bawah Kategori, tambahkan tag <select>:

・Atur attribute name ke

category

.

index.php

Pada tag <select> di bawah Kategori, tambahkan tag <option>:

・Atur attribute value ke

pilih

・Gunakan

Alasan menghubungi kami

sebagai teks tampilan.

index.php

Gunakan iterasi loop foreach dan echo untuk mencetak tag <option> untuk setiap nilai dari array $types.

Gunakan nilai yang dikirim menggunakan method POST, terima dan tampilkan pada halaman sent.php.

sent.php

Di bawah Kategori, gunakan echo untuk print nilai category yang dikirim menggunakan method POST.