



Jurusan Teknologi Informasi - Politeknik Negeri Malang

Jobsheet-4: PHP

Mata Kuliah Desain dan Pemrograman Web

Pengampu: Tim Ajar Desain dan Pemrograman Web

September 2024

Topik

- Pengenalan PHP

Tujuan

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat web statis menggunakan PHP

Pendahuluan

Apa itu PHP ?

PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman server side scripting yang bersifat open source. Sebagai sebuah scripting language, PHP menjalankan instruksi pemrograman saat proses runtime. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. PHP merupakan bahasa pemrograman server-side, maka script dari PHP nantinya akan diproses di server. Jenis server yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, dan LiteSpeed. Selain itu, PHP juga merupakan bahasa pemrograman yang bersifat open source. Pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka.

Mengapa Menggunakan PHP ?

Saat ini, tak kurang dari 78% website di seluruh dunia menggunakan bahasa pemrograman yang diciptakan Rasmus Lerdorf di tahun 1995 ini. Bahkan platform besar seperti Facebook juga menggunakannya. Lalu, apa yang membuat PHP begitu populer? Mengapa menggunakan PHP yang sudah berumur lebih dari dua dekade ini? Berikut beberapa alasannya:

- **Cenderung mudah dipelajari** — dibanding beberapa bahasa pemrograman populer lain, PHP lebih mudah dipelajari.
- **Materi belajar yang melimpah** — umur PHP yang “cukup tua” menyebabkan banyak sekali dokumentasi, panduan, dan komunitas aktif bertebaran di jagat maya. Jadi, tak perlu takut jika Anda mengalami kesulitan.
- **PHP bersifat open-source** — siapapun bisa menggunakan PHP tanpa mengeluarkan biaya sepeserpun.
- **Kecepatan tinggi** — PHP terbukti bisa meningkatkan kecepatan loading dibanding bahasa lain. Misalnya, lebih cepat tiga kali daripada Python pada beberapa kasus.
- **Banyaknya pilihan database** — PHP bisa digunakan di hampir semua jenis database. Mulai dari MySQL, hingga non-relational database seperti Redis.
- **Kompatibilitas yang baik dengan HTML** — script PHP tidak mengganggu HTML sama sekali. Justru mereka berdua saling melengkapi.

- **Fleksibilitas tinggi** — PHP bisa dikombinasikan dengan banyak sekali bahasa pemrograman lain. Sehingga bisa Anda gunakan sesuai kebutuhan.
- **Multi-platform** — PHP bisa Anda gunakan di macam-macam operating system. Mulai dari Windows, Linux, hingga MacOS.
- **Selalu diperbarui** — sejak pertama kali muncul tahun 1995, sekarang PHP sudah berada pada versi 8.2.4.
- **Mendukung layanan cloud** — siapa sangka, walaupun umur PHP hampir dua dekade, tapi ia bisa mendukung layanan cloud dengan skalabilitas yang baik.

Sintaks Dasar PHP

Setiap bahasa pemrograman memiliki aturan coding sendiri. Begitu pula dengan PHP. Semua kode PHP yang ditulis harus disimpan dengan file ekstensi .php. Sintaks dasarnya dibuka dengan `<?php` dan ditutup dengan `?>` sebagai terlihat di contoh berikut:

```
<?php
    echo "Selamat datang";
?>
```

Keterangan:

- **<?php** : kode wajib untuk membuka program PHP.
- **echo** : sebuah perintah untuk menampilkan teks.
- **"Selamat datang"**: teks yang hendak ditampilkan dan ditulis diantara tanda petik dan titik koma.
- **?>** adalah kode untuk mengakhiri PHP dan wajib digunakan saat digabung dengan bahasa pemrograman lain seperti HTML.

Sintaks PHP bersifat **case sensitive**. Jadi, penggunaan huruf besar atau kecil akan turut mempengaruhi output yang diberikan. Sebagai contoh :

```
<?php
    $alamat = "Yogyakarta";
    echo $alamat;
?>
```

Kode di atas akan menghasilkan output: **Yogyakarta** di halaman website.

Namun, jika dituliskan seperti ini:

```
<?php
    $alamat = "Yogyakarta";
    echo $Alamat;
?>
```

Maka akan dihasilkan error. Alasannya, adanya perbedaan antara variabel **\$alamat** dan **\$Alamat**.

Pada PHP, komentar bisa ditulis sebagai penjelasan dari kode. Komentar di PHP menggunakan `//` atau `/*` dan tidak akan dimunculkan sebagai output di browser. Contohnya sebagai berikut:

```
<?php
    // ini contoh penggunaan komentar
    echo "Apa Kabar?";
    /*
    Nah ini juga contoh komentar
    yang ditulis
    lebih dari satu baris
    */
?>
```

Penulisan Kode-Kode PHP

Pada dasarnya, penulisan kode bahasa pemrograman ini terbagi dua :

1. PHP Native

Native adalah penulisan kode PHP dari nol ketika melakukan perancangan sebuah website. PHP Native sering digunakan oleh developer yang memiliki keahlian coding cukup baik atau mereka yang ingin membuat kerangka alur yang unik dengan fungsionalitas tinggi.

2. PHP Framework

Ketika menggunakan framework, developer dapat memanfaatkan kerangka pengelolaan website yang sudah jadi. Artinya, tidak perlu membuatnya dari awal sehingga memudahkan pekerjaan. Framework adalah kerangka kerja yang dapat membantu developer bekerja lebih efisien dan menyelesaikan pengembangan website lebih cepat. Beberapa Framework PHP yang populer digunakan antara lain: CodeIgniter, framework Laravel, Yii, Symfony dan Zend Framework. Jika sudah mahir PHP native, sangat disarankan untuk mencoba beralih ke PHP framework. Itu karena kode pada framework sudah dioptimasi sesuai standar, dari segi kecepatan maupun keamanan.

Variabel Pada PHP

Variabel adalah tempat penyimpanan yang digunakan untuk menyimpan data yang dapat berubah selama jalannya program. Variabel dapat dianggap sebagai wadah yang diisi dengan berbagai jenis data. Berikut adalah beberapa hal yang perlu Anda ketahui tentang variabel dalam PHP:

- **Deklarasi Variabel:** variabel dideklarasikan dengan tanda dollar (\$) diikuti dengan nama variabel. Misalnya: **\$namaVariabel**.
- **Aturan Penamaan Variabel:** nama variabel harus dimulai dengan huruf atau garis bawah (_) dan boleh mengandung huruf, angka, atau garis bawah. Namun, ada perbedaan antara huruf besar dan kecil (case-sensitive). Contoh: **\$nama**, **\$umur**, **\$salamat**.
- **Penugasan Nilai:** nilai dapat diberikan ke variabel dengan menggunakan operator penugasan (=). Contoh: **\$nama = "John";**.
- **Tipe Data:** PHP adalah bahasa pemrograman yang tipe datanya dinamis, artinya tidak perlu menginisialisasi tipe data saat mendeklarasikan variabel. Variabel dapat menyimpan berbagai jenis data, seperti string, angka, boolean, array, dan lain-lain.
- **Mengakses Nilai Variabel:** nilai variabel dapat diakses dengan menggunakan nama variabelnya. Contoh: **\$nama**.

Contoh penerapan script variabel :

- Dalam bentuk string

```
<?php
    $nama = "John";
    $pekerjaan = "Programmer";
    echo "Halo, nama saya $nama dan saya seorang $pekerjaan.";
?>
```

- Dalam bentuk angka

```
<?php
    $umur = 25;
    $tinggi = 175.5;
```

```
echo "Saya berusia $umur tahun dan tinggi saya $tinggi
cm.";
?>
```

Konstanta Pada PHP

Konstanta adalah nilai yang tetap dan tidak dapat diubah selama jalannya program. Mereka sering digunakan untuk menyimpan nilai-nilai yang tidak boleh berubah selama eksekusi program. Berikut adalah beberapa hal tentang konstanta dalam PHP:

- **Deklarasi Konstanta:** Konstanta dideklarasikan menggunakan fungsi **define()** dengan dua parameter: nama konstanta dan nilainya. Contoh: **define("PI", 3.14);**.
- **Aturan Penamaan Konstanta:** Konstanta umumnya ditulis dalam huruf besar dan menggunakan garis bawah (_) sebagai pemisah kata jika nama konstanta terdiri dari beberapa kata. Contoh: **MAX_VALUE**.
- **Tipe Data Konstanta:** Konstanta juga bisa menyimpan berbagai jenis data, seperti string, angka, boolean, atau array. Tapi, sekali konstanta didefinisikan, nilainya tidak dapat diubah selama program berjalan.
- **Mengakses Konstanta:** Anda dapat mengakses konstanta dengan menggunakan nama konstantanya. Contoh: **echo PI;**

Contoh penerapan script konstanta:

```
<?php
define("PI", 3.14159265);
$radius = 5;
$luas_lingkaran = PI * ($radius * $radius);
echo "Luas lingkaran dengan jari-jari $radius adalah $luas_lingkaran.";
?>
```

Praktikum Bagian 1. Variabel dan Konstanta

Langkah	Keterangan
1	Buatlah satu file baru di dalam direktori minggu4 , beri nama variabelkonstanta.php .
2	Tuliskan kode berikut pada file variabelkonstanta.php
3	<pre>1 <?php 2 \$angka1 = 10; 3 \$angka2 = 5; 4 \$hasil = \$angka1 + \$angka2; 5 echo "Hasil penjumlahan \$angka1 dan \$angka2 adalah \$hasil."; 6 ?></pre>
4	Tambahkan isi dari file variabelkonstanta.php dengan kode di bawah ini.
5	<pre>7 \$benar = true; 8 \$salah = false; 9 echo "Variabel benar: \$benar, Variabel salah: \$salah"; 10 ?></pre>
6	Tambahkan isi dari file variabelkonstanta.php dengan kode di bawah ini.

7	<pre> 11 // Mendefinisikan konstanta untuk nilai tetap 12 define("NAMA_SITUS", "WebsiteKu.com"); 13 define("TAHUN_PENDIRIAN", 2023); 14 15 echo "Selamat datang di " . NAMA_SITUS . ", situs yang didirikan pada tahun " . TAHUN_PENDIRIAN . "."; 16 ?> </pre>
8	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/variabelkonstanta</code> .
9	Apa yang Anda pahami dari penggunaan variabel pada file tersebut? Catat pemahaman Anda di bawah ini. (Pertanyaan No. 1)

Type Data Pada PHP

Tipe data adalah klasifikasi jenis data atau bentuk dari suatu data. Ada berbagai macam tipe data mulai dari tipe data asli dan tipe data buatan. PHP mendukung setidaknya delapan tipe data skalar. Delapan tipe data tersebut adalah:

Type Data	Keterangan
Integer	Berisi bilangan bulat.
Float	Berisi bilangan desimal.
Boolean	Berisi 2 nilai saja: true dan false.
String	Berisi data teks yang diapit oleh tanda ' ' atau " ".
Array	Berisi himpunan data.
Object	Digunakan untuk membuat objek dari kelas yang telah didefinisikan sebelumnya.
NULL	Digunakan untuk menyatakan bahwa suatu variabel tidak memiliki nilai.
Callable	Digunakan untuk menyimpan referensi ke fungsi atau metode yang dapat dipanggil.

Praktikum 2 : Penggunaan Tipe Data

Langkah	Keterangan
1	Buat file baru bernama <code>tipedata.php</code> . Ketikkan kode pada langkah 2 di dalam <code>tipedata.php</code>

2	<pre> 1 <?php 2 \$a = 10; 3 \$b = 5; 4 \$c = \$a + 5; 5 \$d = \$b + (10 * 5); 6 \$e = \$d - \$c; 7 8 echo "Variabel a: {\$a}
"; 9 echo "Variabel b: {\$b}
"; 10 echo "Variabel c: {\$c}
"; 11 echo "Variabel d: {\$d}
"; 12 echo "Variabel e: {\$e}
"; 13 14 var_dump(\$e); 15 ?> </pre>
3	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/tipedata.php</code>
4	Jelaskan yang Anda pahami dari kode tersebut di bawah ini. (Pertanyaan No. 2)
5	<p>Tambahkan isi dari file <code>tipedata.php</code> dengan kode di bawah ini.</p> <pre> 16 \$nilaiMatematika = 5.1; 17 \$nilaiIPA = 6.7; 18 \$nilaiBahasaIndonesia = 9.3; 19 20 \$rataRata = (\$nilaiMatematika + \$nilaiIPA + \$nilaiBahasaIndonesia) / 3; 21 22 echo "Matematika: {\$nilaiMatematika}
"; 23 echo "IPA: {\$nilaiIPA}
"; 24 echo "Bahasa Indonesia: {\$nilaiBahasaIndonesia}
"; 25 echo "Rata-rata: {\$rataRata}
"; 26 27 var_dump(\$rataRata); 28 ?> </pre>
6	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/tipedata.php</code>
7	Jelaskan yang Anda pahami dari kode tersebut di bawah ini. (Pertanyaan No. 3)
8	Tambahkan isi file <code>tipedata.php</code> dengan kode di bawah ini.

	<pre> 29 \$apakahSiswaLulus = true; 30 \$apakahSiswaSudahUjian = false; 31 32 var_dump(\$apakahSiswaLulus); 33 echo "
"; 34 var_dump(\$apakahSiswaSudahUjian); 35 ?> </pre>
9	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/tipedata.php</code>
10	Jelaskan yang Anda pahami dari kode tersebut di bawah ini. (Pertanyaan No. 4)
11	<p>Tambahkan isi file <code>tipedata.php</code> dengan kode di bawah ini.</p> <pre> 36 \$namaDepan = "Ibnu"; 37 \$namaBelakang = 'Jakaria'; 38 39 \$namaLengkap = "{\$namaDepan} {\$namaBelakang}"; 40 \$namaLengkap2 = \$namaDepan . ' ' . \$namaBelakang; 41 42 echo "Nama Depan: {\$namaDepan}
"; 43 echo 'Nama Belakang: ' . \$namaBelakang . '
'; 44 45 echo \$namaLengkap; 46 ?> </pre>
12	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/tipedata.php</code>
13	Jelaskan yang Anda pahami dari kode tersebut di bawah ini. (Pertanyaan No. 5)
14	<p>Tambahkan isi file <code>tipedata.php</code> dengan kode di bawah ini.</p> <pre> 47 \$listMahasiswa = ["Wahid Abdullah", "Elmo Bachtiar", "Lendis Fabri"]; 48 echo \$listMahasiswa[0]; 49 ?> </pre>
15	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/tipedata.php</code>
16	Jelaskan yang Anda pahami dari kode tersebut di bawah ini. (Pertanyaan No. 6)

Operator Pada PHP

Operator dalam pemrograman adalah simbol atau tanda yang digunakan untuk melakukan operasi pada data. Operator digunakan untuk melakukan berbagai tindakan seperti operasi matematika, perbandingan, logika, dan lainnya pada nilai-nilai yang ada dalam program. Kita menggunakan operator dalam bahasa pemrograman PHP ketika perlu melakukan operasi dasar seperti manipulasi data, perbandingan nilai, evaluasi kondisi, dan berbagai tindakan lainnya pada nilai-nilai yang ada dalam program. Operator memungkinkan untuk mengubah, memanipulasi, atau mengambil keputusan berdasarkan nilai-nilai tersebut.

Jenis-Jenis Operator Pada PHP

Berikut adalah tabel yang berisi daftar beberapa operator yang umum digunakan dalam PHP

Jenis Operator	Simbol	Deskripsi dan Contoh Penggunaan
Aritmatika	+	Penjumlahan: \$hasil = \$angka1 + \$angka2;
	-	Pengurangan: \$hasil = \$angka1 - \$angka2;
	*	Perkalian: \$hasil = \$angka1 * \$angka2;
	/	Pembagian: \$hasil = \$angka1 / \$angka2;
	%	Sisa Bagi: \$sis = \$angka1 % \$angka2;
Pembandingan	==	Sama dengan: \$hasil = \$nilai1 == \$nilai2;
	!=	Tidak sama dengan: \$hasil = \$nilai1 != \$nilai2;
	<	Lebih kecil dari: \$hasil = \$nilai1 < \$nilai2;
	>	Lebih besar dari: \$hasil = \$nilai1 > \$nilai2;
	<=	Lebih kecil dari atau sama dengan: \$hasil = \$nilai1 <= \$nilai2;
Logika	>=	Lebih besar dari atau sama dengan: \$hasil = \$nilai1 >= \$nilai2;
	&& / and	AND (dan): \$hasil = \$a && \$b;
	/ or	OR (atau): \$hasil = \$a \$b;
Penugasan	! / not	NOT (negasi): \$hasil = !\$a;
	=	Penugasan biasa: \$a = \$b;
	+=	Penugasan dengan penjumlahan: \$a += \$b;
	-=	Penugasan dengan pengurangan: \$a -= \$b;
	*=	Penugasan dengan perkalian: \$a *= \$b;
	/=	Penugasan dengan pembagian: \$a /= \$b;
	%=	Penugasan dengan sisa bagi: \$a %= \$b;
Increment/ Decrement	++	Increment (menambah 1): \$a++;
	--	Decrement (mengurangi 1): \$a--;
Bitwise	&	AND bitwise: \$hasil = \$a & \$b;
		OR bitwise: \$hasil = \$a \$b;
	^	XOR bitwise: \$hasil = \$a ^ \$b;
	~	NOT bitwise: \$hasil = ~\$a;
Pemangkatan	**	Pangkat: \$hasil = \$angka ** \$pangkat;
Identik	===	Identik (nilai dan tipe data sama): \$hasil = \$a === \$b;
	!==	Tidak Identik (nilai atau tipe data tidak sama): \$hasil = \$a !== \$b;
Ternary	? :	Operator kondisional (ternary): \$hasil = (\$nilai > 10) ? "Lebih besar" : "Lebih kecil";
Array	[]	Akses elemen array: \$nilai = \$array[0];
	+	Penggabungan array: \$array3 = \$array1 + \$array2;

Praktikum Bagian 3 : Penggunaan Operator PHP

Langkah	Keterangan
1	Buat satu file baru bernama <code>operator.php</code> . Ketikkan kode pada langkah 2 di dalam <code>operator.php</code>
2	<pre> 1 <?php 2 \$a = 10; 3 \$b = 5; 4 5 \$hasilTambah = \$a + \$b; 6 \$hasilKurang = \$a - \$b; 7 \$hasilKali = \$a * \$b; 8 \$hasilBagi = \$a / \$b; 9 \$sisasBagi = \$a % \$b; 10 \$pangkat = \$a ** \$b; </pre>
3	Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. (Pertanyaan No. 7)
4	Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/minggu4/operator.php</code> . Amati yang terjadi dan jelaskan yang Anda pahami. (Pertanyaan No. 8)
5	<p>Tambahkan kode berikut pada <code>operator.php</code></p> <pre> 12 \$hasilSama = \$a == \$b; 13 \$hasilTidakSama = \$a != \$b; 14 \$hasilLebihKecil = \$a < \$b; 15 \$hasilLebihBesar = \$a > \$b; 16 \$hasilLebihKecilSama = \$a <= \$b; 17 \$hasilLebihBesarSama = \$a >= \$b; 18 </pre>
6	Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/operator.php</code> Amati yang terjadi dan jelaskan yang Anda pahami. (Pertanyaan No. 9)
7	Tambahkan kode pada langkah 8 pada <code>operator.php</code>
8	<pre> 19 \$hasilAnd = \$a && \$b; 20 \$hasilOr = \$a \$b; 21 \$hasilNotA = !\$a; 22 \$hasilNotB = !\$b; </pre>

9	Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/operator.php</code> Amati yang terjadi dan jelaskan yang Anda pahami. (Pertanyaan No. 10)
10	Tambahkan kode pada langkah 11 pada <code>operator.php</code>
11	<pre> 24 \$a += \$b; 25 \$a -= \$b; 26 \$a *= \$b; 27 \$a /= \$b; 28 \$a %= \$b; </pre>
12	Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/operator.php</code> Amati yang terjadi dan jelaskan yang Anda pahami. (Pertanyaan No. 11)
13	Tambahkan kode pada langkah 14 pada <code>operator.php</code>
14	<pre> 30 \$hasilIdentik = \$a === \$b; 31 \$hasilTidakIdentik = \$a !== \$b; </pre>
15	Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/operator.php</code> Amati yang terjadi dan jelaskan yang Anda pahami. (Pertanyaan No. 12)
16	Soal: Sebuah restoran memiliki 45 kursi di dalamnya. Pada suatu malam, 28 kursi telah ditempati oleh pelanggan. Berapa persen kursi yang masih kosong di restoran tersebut?
17	Buat file dengan nama <code>latihanoperator.php</code> . Tuliskan kode program untuk langkah 16 dan tampilkan hasilnya di bawah ini beserta kode programnya (Pertanyaan No. 13)

Struktur Kontrol Pada PHP

Pada dasarnya, pemrograman adalah tentang mengendalikan alur eksekusi program. Struktur kontrol adalah alat yang digunakan untuk memutuskan bagaimana kode akan dieksekusi berdasarkan kondisi-kondisi tertentu.

Pernyataan Kondisional

Pernyataan kondisional digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Ada beberapa jenis pernyataan kondisional dalam PHP:

- **if:** Digunakan untuk menjalankan blok kode jika kondisi tertentu benar.

```
if ($nilai > 90) {  
    echo "Nilai Anda sangat baik!";  
}
```

- **else:** Digunakan bersamaan dengan **if** untuk menjalankan blok kode jika kondisi tidak benar.

```
if ($nilai > 90) {  
    echo "Nilai Anda sangat baik!";  
} else {  
    echo "Anda perlu belajar lebih keras."  
}
```

- **elseif:** Digunakan untuk menambahkan kondisi tambahan setelah **if**.

```
if ($nilai > 90) {  
    echo "A+";  
} elseif ($nilai > 80) {  
    echo "A";  
} else {  
    echo "Belajar lebih keras."  
}
```

- **switch:** Digunakan untuk mengevaluasi ekspresi dan menjalankan kode berdasarkan nilai yang cocok.

```
switch ($hari) {  
    case "Senin":  
        echo "Hari kerja."  
        break;  
    case "Sabtu":  
    case "Minggu":  
        echo "Akhir pekan."  
        break;  
    default:  
        echo "Hari tidak valid."  
}
```

Perulangan

Perulangan memungkinkan Anda untuk mengulang kode beberapa kali berdasarkan kondisi tertentu. Ada tiga jenis perulangan yang paling umum digunakan dalam PHP:

- **for:** Digunakan untuk mengulang kode sejumlah tertentu kali.

```
for ($i = 1; $i <= 5; $i++) {
    echo "Iterasi ke-$i <br>";
}
```

- **while**: Digunakan untuk mengulang kode selama kondisi tertentu benar.

```
$counter = 1;
while ($counter <= 5) {
    echo "Iterasi ke-$counter <br>";
    $counter++;
}
```

- **do-while**: Mirip dengan **while**, tetapi akan menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi.

```
$counter = 1;
do {
    echo "Iterasi ke-$counter <br>";
    $counter++;
} while ($counter <= 5);
```

Pernyataan Pengendalian Aliran

Pernyataan ini memungkinkan Anda mengontrol aliran eksekusi program dengan cara tertentu:

- **break**: Digunakan untuk menghentikan perulangan atau switch saat ini.

```
for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        break; // Menghentikan perulangan saat $i = 5
    }
    echo "Iterasi ke-$i <br>";
}
```

```
Iterasi ke-1
Iterasi ke-2
Iterasi ke-3
Iterasi ke-4
```

- **continue**: Digunakan untuk melanjutkan iterasi berikutnya dalam perulangan.

```
for ($i = 1; $i <= 5; $i++) {
    if ($i == 3) {
        continue; // Melanjutkan ke iterasi berikutnya saat $i = 3
    }
    echo "Iterasi ke-$i <br>";
}
```

```
Iterasi ke-1
Iterasi ke-2
Iterasi ke-4
Iterasi ke-5
```

Praktikum Bagian 4 : Penggunaan Struktur Kontrol Pada PHP

Langkah	Keterangan
1	Buat satu file baru bernama strukturkontrol.php . Ketikkan kode pada langkah 2 di dalam strukturkontrol.php

2	<pre> 1 <?php 2 \$nilaiNumerik = 92; 3 4 if (\$nilaiNumerik >= 90 && \$nilaiNumerik <= 100) { 5 echo "Nilai huruf: A"; 6 } elseif (\$nilaiNumerik >= 80 && \$nilaiNumerik < 90) { 7 echo "Nilai huruf: B"; 8 } elseif (\$nilaiNumerik >= 70 && \$nilaiNumerik < 80) { 9 echo "Nilai huruf: C"; 10 } elseif (\$nilaiNumerik < 70) { 11 echo "Nilai huruf: D"; 12 } 13 ?> </pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh localhost/minggu4/strukturkontrol.php
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 14)
5	Tambahkan kode pada langkah 6 di strukturkontrol.php
6	<pre> \$jarakSaatIni = 0; \$jarakTarget = 500; \$peningkatanHarian = 30; \$hari = 0; while (\$jarakSaatIni < \$jarakTarget) { \$jarakSaatIni += \$peningkatanHarian; \$hari++; } echo "Atlet tersebut memerlukan \$hari hari untuk mencapai jarak 500 kilometer."; </pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/minggu4/strukturkontrol.php
8	Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 15)
9	Tambahkan kode pada langkah 9 di dalam strukturkontrol.php
10	<pre> \$jumlahLahan = 10; \$tanamanPerLahan = 5; \$buahPerTanaman = 10; \$jumlahBuah = 0; for (\$i = 1; \$i <= \$jumlahLahan; \$i++) { \$jumlahBuah += (\$tanamanPerLahan * \$buahPerTanaman); } echo "Jumlah buah yang akan dipanen adalah: \$jumlahBuah"; </pre>

11	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/strukturkontrol.php</code>
12	Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 16)
13	Tambahkan kode pada langkah 14 di dalam <code>strukturkontrol.php</code>
14	<pre> \$skorUjian = [85, 92, 78, 96, 88]; \$totalSkor = 0; foreach (\$skorUjian as \$skor) { \$totalSkor += \$skor; } echo "Total skor ujian adalah: \$totalSkor"; </pre>
15	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/strukturkontrol.php</code>
16	Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 17)
17	Tambahkan kode pada langkah 18 di dalam <code>strukturkontrol.php</code>
18	<pre> \$nilaiSiswa = [85, 92, 58, 64, 90, 55, 88, 79, 70, 96]; foreach (\$nilaiSiswa as \$nilai) { if (\$nilai < 60) { echo "Nilai: \$nilai (Tidak lulus)
"; continue; } echo "Nilai: \$nilai (Lulus)
"; } </pre>
19	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/strukturkontrol.php</code>
20	Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 18)
21	<p>Soal: Ada seorang guru ingin menghitung total nilai dari 10 siswa dalam ujian matematika. Guru ini ingin mengabaikan dua nilai tertinggi dan dua nilai terendah. Bantu guru ini menghitung total nilai yang akan digunakan untuk menentukan nilai rata-rata setelah mengabaikan nilai tertinggi dan terendah. Berikut daftar nilai dari 10 siswa (85, 92, 78, 64, 90, 75, 88, 79, 70, 96)</p>

22	Buat file dengan nama <code>kontrol1.php</code> . Buat kode program untuk langkah 21 dan tampilkan hasilnya di bawah ini beserta kode programnya (Pertanyaan No. 19)
23	Ada soal cerita : Seorang pelanggan ingin membeli sebuah produk dengan harga Rp 120.000. Toko tersebut menawarkan diskon sebesar 20% untuk pembelian di atas Rp 100.000. Bantu pelanggan ini untuk menghitung harga yang harus dibayar setelah mendapatkan diskon.
24	Buat file dengan nama <code>kontrol2.php</code> . Buat kode program untuk langkah 23 dan tampilkan hasilnya di bawah ini beserta kode programnya (Pertanyaan No. 20)
25	Soal: Seorang pemain game ingin menghitung total skor mereka dalam permainan. Mereka mendapatkan skor berdasarkan poin yang mereka kumpulkan. Jika mereka memiliki lebih dari 500 poin, maka mereka akan mendapatkan hadiah tambahan. Buat tampilan baris pertama "Total skor pemain adalah: (poin)". Dan baris kedua "Apakah pemain mendapatkan hadiah tambahan? (YA/TIDAK)"
26	Buat file dengan nama <code>kontrol3.php</code> . Buat kode program untuk langkah 25 tampilkan hasilnya di bawah ini beserta kode programnya (Pertanyaan No. 21)

Praktikum Bagian 5 : Penggunaan Array Pada PHP

Array Satu Dimensi

Array satu dimensi adalah struktur data yang digunakan untuk menyimpan sejumlah nilai dalam satu variabel. Anda dapat mendeklarasikan array satu dimensi dalam PHP menggunakan beberapa metode:

Metode 1: Menggunakan `array()`

```
$buah = array("apel", "jeruk", "pisang", "mangga");
```

Metode 2: Menggunakan tanda kurung siku `[]` (Mulai dari PHP 5.4+)

```
$buah = ["apel", "jeruk", "pisang", "mangga"];
```

Untuk mengakses nilai dalam array digunakan indeks (mulai dari 0 untuk indeks pertama):

```
echo $buah[0]; // Output: apel
echo $buah[2]; // Output: pisang
```

Anda dapat menambahkan nilai ke dalam array menggunakan indeks tertentu atau dengan metode `[]` untuk menambahkan nilai baru ke akhir array:

```
$buah[] = "anggur"; // Menambahkan "anggur" ke akhir array
$buah[4] = "durian"; // Menambahkan "durian" pada indeks 4
```

Anda dapat mengubah nilai dalam array dengan mengakses indeksnya dan menggantinya:

```
$buah[1] = "semangka"; // Mengubah nilai "jeruk" menjadi "semangka"
```

Anda dapat melakukan iterasi (perulangan) melalui semua elemen dalam array menggunakan loop **for**, **foreach**, atau **while**. Berikut contohnya dengan **foreach**:

```
foreach ($buah as $item) {
    echo $item . "<br>";
}
```

Array Dua Dimensi

Array dua dimensi adalah struktur data yang digunakan untuk menyimpan data dalam bentuk tabel dua dimensi, yang mirip dengan baris dan kolom dalam spreadsheet. Berikut adalah penjelasan singkat tentang array dua dimensi dan contoh penerapannya dalam PHP:

1. Deklarasi Array Dua Dimensi

Anda dapat mendeklarasikan array dua dimensi dalam PHP dengan menyusun array satu dimensi dalam array lainnya:

```
$matrix = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
```

Anda juga dapat menggunakan sintaks tanda kurung siku [[]]:

```
$matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
];
```

2. Mengakses Nilai dalam Array Dua Dimensi

Anda dapat mengakses nilai dalam array dua dimensi dengan menggunakan dua indeks, satu untuk baris dan satu lagi untuk kolom:

```
echo $matrix[0][0]; // Output: 1 (baris 1, kolom 1)
echo $matrix[1][2]; // Output: 6 (baris 2, kolom 3)
```

3. Menambahkan dan Mengubah Nilai dalam Array Dua Dimensi

Anda dapat menambahkan atau mengubah nilai dalam array dua dimensi dengan mengakses elemen menggunakan indeks:


```
$matrix[1][1] = 10; // Mengubah nilai pada baris 2, kolom 2 menjadi 10
$matrix[2][] = 11; // Menambahkan nilai 11 ke akhir array pada baris 3
```

4. Iterasi Melalui Array Dua Dimensi

Anda dapat melakukan iterasi melalui semua elemen dalam array dua dimensi menggunakan nested loop, seperti loop **for**:

```
for ($i = 0; $i < count($matrix); $i++) {
    for ($j = 0; $j < count($matrix[$i]); $j++) {
        echo $matrix[$i][$j] . " ";
    }
    echo "<br>";
}
```

Ini akan mencetak seluruh elemen dalam array dua dimensi, menghasilkan tampilan seperti tabel.

1	2	3
4	5	6
7	8	9

Array Multidimensi

Array multidimensi adalah struktur data yang memungkinkan Anda untuk menyimpan data dalam lebih dari dua dimensi. Ini adalah pengembangan dari array dua dimensi yang telah dijelaskan sebelumnya.

1. Deklarasi Array Multidimensi Tiga Dimensi

Anda dapat mendeklarasikan array multidimensi tiga dimensi dengan menambahkan array dalam array dalam array:

```
$threeDArray = array(
    array(
        array(1, 2, 3),
        array(4, 5, 6)
    ),
    array(
        array(7, 8, 9),
        array(10, 11, 12)
    )
);
```

Ini adalah contoh array tiga dimensi. Anda juga dapat menggunakan sintaks tanda kurung siku [][][]:

```
$threeDArray = [
    [
        [1, 2, 3],
        [4, 5, 6]
    ],
    [
        [7, 8, 9],
        [10, 11, 12]
    ]
];
```

2. Mengakses Nilai dalam Array Multidimensi Tiga Dimensi

Anda dapat mengakses nilai dalam array multidimensi tiga dimensi dengan menggunakan tiga indeks, masing-masing untuk tingkat dimensi yang sesuai:

```
echo $threeDArray[0][0][0]; // Output: 1 (dimensi 1, dimensi 2, elemen pertama)
echo $threeDArray[1][1][2]; // Output: 12 (dimensi 2, dimensi 2, elemen ketiga)
```

3. Menambahkan dan Mengubah Nilai dalam Array Multidimensi Tiga Dimensi

Anda dapat menambahkan atau mengubah nilai dalam array multidimensi tiga dimensi dengan mengakses elemen menggunakan indeks:

```
$threeDArray[0][1][1] = 15; // Mengubah nilai pada dimensi 1, dimensi 3, elemen kedua
$threeDArray[1][0][] = 13; // Menambahkan nilai 13 ke akhir array pada dimensi 1
```

4. Iterasi Melalui Array Multidimensi Tiga Dimensi

Untuk mengiterasi melalui array multidimensi tiga dimensi, Anda perlu menggunakan tiga tingkat nested loop:

```
foreach ($threeDArray as $dimensi1) {
    foreach ($dimensi1 as $dimensi2) {
        foreach ($dimensi2 as $elemen) {
            echo $elemen . " ";
        }
        echo "<br>";
    }
    echo "<br>";
}
```

Dalam contoh di atas, kita menggunakan tiga tingkat nested loop untuk mengakses semua elemen dalam array tiga dimensi dan mencetaknya.

```
1 2 3
4 5 6

7 8 9
10 11 12
```

Array Asosiatif

Array asosiatif (associative array) adalah tipe array di PHP yang menggunakan pasangan "kunci" (key) dan "nilai" (value) untuk mengorganisir data. Berikut adalah penjelasan dan contoh penerapan array asosiatif dalam PHP:

1. Deklarasi Array Asosiatif

Anda dapat mendeklarasikan array asosiatif dengan menggunakan sintaks **array()** atau tanda kurung siku []. Setiap elemen array asosiatif memiliki kunci (key) yang berpasangan dengan nilai (value)nya. Berikut adalah contoh penggunaan array asosiatif dalam PHP:

```
// Menggunakan sintaks array()
$student = array(
    "nama" => "John",
    "usia" => 20,
    "kelas" => "12A"
);

// Menggunakan sintaks tanda kurung siku []
$student = [
    "nama" => "John",
    "usia" => 20,
    "kelas" => "12A"
];
```

2. Mengakses Nilai dalam Array Asosiatif

Anda dapat mengakses nilai dalam array asosiatif dengan menggunakan kunci yang sesuai:

```
echo $student["nama"]; // Output: John
echo $student["usia"]; // Output: 20
echo $student["kelas"]; // Output: 12A
```

3. Menambahkan dan Mengubah Nilai dalam Array Asosiatif

Anda dapat menambahkan atau mengubah nilai dalam array asosiatif dengan mengakses elemen menggunakan kunci:

```
$student["nilai"] = 95; // Menambahkan nilai baru ke dalam array
$student["usia"] = 21; // Mengubah nilai usia
```

4. Iterasi Melalui Array Asosiatif

```
foreach ($student as $key => $value) {
    echo "$key: $value <br>";
}
```

Output dari loop di atas akan mencetak semua pasangan kunci-nilai dalam array asosiatif.

```
nama: John
usia: 21
kelas: 12A
nilai: 95
```

Langkah	Keterangan
1	Buat file baru bernama <code>array.php</code> . Ketikkan kode pada langkah 2 di dalam <code>array.php</code>
2	<pre> \$nilaiSiswa = [85, 92, 78, 64, 90, 55, 88, 79, 70, 96]; \$nilaiLulus = []; foreach (\$nilaiSiswa as \$nilai) { if (\$nilai >= 70) { \$nilaiLulus[] = \$nilai; } } echo "Daftar nilai siswa yang lulus: " . implode(', ', \$nilaiLulus); </pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/array.php</code>
4	Catat di sini apa yang anda amati dari penambahan kode di atas. (Pertanyaan No. 22)
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>array.php</code>
6	<pre> \$daftarKaryawan = [['Alice', 7], ['Bob', 3], ['Charlie', 9], ['David', 5], ['Eva', 6],]; \$karyawanPengalamanLimaTahun = []; foreach (\$daftarKaryawan as \$karyawan) { if (\$karyawan[1] > 5) { \$karyawanPengalamanLimaTahun[] = \$karyawan[0]; } } echo "Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: " . implode(', ', \$karyawanPengalamanLimaTahun); </pre>
7	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/minggu4/array.php</code>
8	Catat di sini apa yang Anda amati dari penambahan kode di atas. (Pertanyaan No. 23)
9	Ketikkan kode tambahan pada langkah 10 di dalam <code>array.php</code>

10	<pre> \$daftarNilai = ['Matematika' => [['Alice', 85], ['Bob', 92], ['Charlie', 78],], 'Fisika' => [['Alice', 90], ['Bob', 88], ['Charlie', 75],], 'Kimia' => [['Alice', 92], ['Bob', 80], ['Charlie', 85],],]; \$mataKuliah = 'Fisika'; echo "Daftar nilai mahasiswa dalam mata kuliah \$mataKuliah:
"; foreach (\$daftarNilai[\$mataKuliah] as \$nilai) { echo "Nama: {\$nilai[0]}, Nilai: {\$nilai[1]}
"; } </pre>
11	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/minggu4/array.php</p>
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 24)</p>
13	<p>Soal: Seorang guru ingin mencetak daftar nilai siswa dalam ujian matematika. Guru tersebut memiliki data setiap siswa terdiri dari nama dan nilai. Bantu guru ini mencetak daftar nilai siswa yang mencapai nilai di atas rata-rata kelas. Dengan ketentuan nama dan nilai siswa Alice dapat 85, Bob dapat 92, Charlie dapat 78, David dapat 64, Eva dapat 90</p>
14	<p>Buat kode untuk langkah 13 dengan array dua dimensi dan tampilkan hasilnya di bawah ini beserta kode programnya (Pertanyaan No. 25)</p>