



Nama : Lentina Abrila Putri
Nim : 2241760120
Prodi : SIB 2C

Topik

- Pengenalan PHP

Tujuan

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat web statis menggunakan PHP

Perhatian

Jobsheet ini harus dikerjakan step-by-step sesuai langkah-langkah praktikum yang sudah diberikan. Soal dapat dijawab langsung di dalam kolom yang disediakan dengan menggunakan PDF Editor.

Pendahuluan

Apakah 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. Sintaks dasarnya dibuka dengan `<?php` dan ditutup dengan `?>` sebagai terlihat di contoh berikut:

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

Inilah penjelasan kode tersebut:

- `<?php` ini adalah kode wajib untuk membuka program PHP.
- `Echo` adalah 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;
?>
```

Anda akan mendapati tampilan error. Alasannya, adanya perbedaan antara variabel **\$alamat** dan **\$Alamat**. Pada PHP, Anda juga bisa menuliskan komentar sebagai penjelasan dari kode yang ditulis. 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
*/
?>
```

Semua kode PHP yang ditulis harus disimpan dengan file ekstensi **.php**.

Penulisan Kode-Kode PHP

Anda baru saja belajar contoh 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 Anda 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. Anda dapat menganggapnya sebagai wadah yang dapat Anda isi dengan berbagai jenis data. Berikut adalah beberapa hal yang perlu Anda ketahui tentang variabel dalam PHP:

- **Deklarasi Variabel:** Anda bisa mendeklarasikan variabel dengan mengawali nama variabel 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:** Anda dapat memberikan nilai ke variabel dengan menggunakan operator penugasan (=). Contoh: **\$nama = "John";**.
- **Tipe Data:** PHP adalah bahasa pemrograman yang tipe datanya dinamis, artinya Anda tidak perlu menginisialisasi tipe data saat mendeklarasikan variabel. Variabel dapat menyimpan berbagai jenis data, seperti string, angka, boolean, array, dan banyak lagi.
- **Mengakses Nilai Variabel:** Anda dapat mengakses nilai variabel dengan menggunakan nama variabelnya. Contoh: **\$nama**.

Contoh penerapan script variabel :

- Dalam bentuk string

```
<?php
$name = "John";
$pekerjaan = "Programmer";
echo "Halo, nama saya $name 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 yang perlu Anda ketahui 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:

```
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 dasarWeb, beri nama variabel_konstanta.php.
2	Ketikkan ke dalam file variabel_konstanta.php tersebut kode di bawah ini.
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 variabel_konstanta.php tersebut 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 variabel_konstanta.php tersebut 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	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/variabel_konstanta buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah</p> <p>Hasilnya:</p>

	<p>Hasil penjumlahan 10 dan 5 adalah 15. Variabel benar: 1 Variabel salah: Selamat datang di WebsiteKu.com, situs yang didirikan pada tahun 2023.</p>
9	<p>Apa yang anda pahami dari penggunaan variable pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1)</p> <p>Penjelasan:</p> <ul style="list-style-type: none"> • Kode program yang pertama adalah kode yang melakukan penjumlahan antara variabel \$angka1 dan \$angka2. Dimana \$angka1 adalah 10 dan \$angka2 adalah 5, sehingga jika dilakukan penjumlahan akan menghasilkan hasil 15. • Kode yang diberikan adalah kode PHP yang menetapkan nilai ke variabel \$benar dan \$salah. Nilai true direpresentasikan sebagai 1 dalam konteks percetakan, sedangkan nilai false direpresentasikan sebagai string kosong. • Kode tersebut mendefinisikan 2 konstanta menggunakan fungsi define() dalam kode PHP, yaitu NAMA_SITUS dan TAHUN_PENDIRIAN. Kemudian kode tersebut mencetak pesan selamat datang.

Tipe Data Pada PHP

Tipe data adalah klasifikasi jenis data atau bentukan dari suatu data. Ia menjelaskan suatu data: dari jenis apakah ia tersusun? Apakah bilangan riil? Atau kah bilangan pecahan? Atau kah ia data yang tersusun dari bentukan karakter? Intinya tipe data adalah klasifikasi jenis dari data yang kita ingin simpan dalam sebuah variabel.

Macam-Macam Tipe Data Dalam PHP

Ada berbagai macam tipe data: mulai dari tipe data asli dan tipe data buatan. Untuk PHP sendiri, ia mendukung setidaknya 8 tipe data skalar. 8 tipe data tersebut adalah:

Tipe 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 satu file baru bernama tipe_data.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam tipe_data.php

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	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <pre> Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 40 int(40) </pre>
4	<p>Tambahkan isi dari file tipe_data.php tersebut dengan kode di bawah ini.</p>
5	<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	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p>Hasilnya:</p> <pre> Matematika: 5.1 IPA: 6.7 Bahasa Indonesia: 9.3 Rata-rata: 7.0333333333333 float(7.03333333333334) </pre>
7	<p>Tambahkan isi dari file tipe_data.php tersebut dengan kode di bawah ini.</p>

8	<pre> 29 \$apakahSiswaLulus = true; 30 \$apakahSiswaSudahUjian = false; 31 32 var_dump(\$apakahSiswaLulus); 33 echo "
"; 34 var_dump(\$apakahSiswaSudahUjian); 35 ?> </pre>
9	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p>Hasilnya:</p> <pre> bool(true) bool(false) </pre>
10	<p>Tambahkan isi dari file tipe_data.php tersebut dengan kode di bawah ini.</p>
11	<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	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p>Hasilnya:</p> <pre> ----- Nama Depan: Ibnu Nama Belakang: Jakarta Ibnu Jakarta </pre>
13	<p>Tambahkan isi dari file tipe_data.php tersebut dengan kode di bawah ini.</p>
14	<pre> 47 \$listMahasiswa = ["Wahid Abdullah", "Elmo Bachtiar", "Lendis Fabri"]; 48 echo \$listMahasiswa[0]; 49 ?> </pre>
15	<p>Rapikan hasil supaya ada jarak per tipe data. Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p>Hasilnya:</p>

	<p>Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 40</p> <p>int(40) Matematika: 5.1 IPA: 6.7 Bahasa Indonesia: 9.3 Rata-rata: 7.0333333333333 float(7.03333333333334)</p> <p>bool(true) bool(false)</p> <p>Nama Depan: Ibnu Nama Belakang: Jakarta Ibnu Jakarta</p> <p>Wahid Abdullah</p>
16	<p>Apa yang anda pahami dari penggunaan tipe data pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2)</p> <p>Penjelasan:</p> <ul style="list-style-type: none"> • Kode tersebut melakukan operasi aritmatika pada variabel-variabel yang telah ditentukan sebelumnya, kemudian mencetak nilai dari masing-masing variabel serta melakukan tindakan <code>var_dump()</code> pada variabel <code>\$e</code>. • Kode tersebut melakukan perhitungan rata-rata dari tiga nilai, yaitu matematika, ipa, dan bahasa indonesia, kemudian mencetak nilai masing-masing variabel serta rata-rata nilai tersebut. Dan akhirnya menampilkan informasi tipe data dan nilai dari variabel rata-rata menggunakan <code>var_dump()</code>. • Kode ini menjelaskan mengenai 2 variabel boolean, <code>\$apakahSiswaLulus</code> dan <code>\$apakahSiswaSudahUjian</code>, yang masing-masing memiliki nilai <code>true</code> dan <code>false</code>. Kemudian, kode tersebut menggunakan fungsi <code>var_dump()</code> untuk menampilkan informasi tipe data dan nilai dari masing-masing variabel. • Kode ini menjelaskan 2 variabel string, <code>\$namaDepan</code> dan <code>\$namaBelakang</code>, yang masing-masing menyimpan nama depan dan nama belakang. Kemudian kode tersebut menggunakan kedua variabel tersebut untuk membentuk nama lengkap dan mencetaknya dalam 2 cara yang berbeda. • Kode selanjutnya menjelaskan sebuah array dengan nama <code>\$listMahasiswa</code> yang berisi daftar nama mahasiswa. Kemudian, kode tersebut mencetak nama mahasiswa pertama.

Operator Pada PHP

Operator dalam pemrograman adalah simbol atau tanda yang digunakan untuk melakukan operasi pada data. Operator memungkinkan Anda 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 kita perlu melakukan operasi dasar seperti manipulasi data, perbandingan nilai, evaluasi kondisi, dan berbagai tindakan lainnya pada nilai-nilai yang ada dalam program yang kita kembangkan. Operator memungkinkan kita 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: <code>\$hasil = \$angka1 + \$angka2;</code>

	—	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;
	>=	Lebih besar dari atau sama dengan: \$hasil = \$nilai1 >= \$nilai2;
Logika	&& / and	AND (dan): \$hasil = \$a && \$b;
	/ or	OR (atau): \$hasil = \$a \$b;
	! / not	NOT (negasi): \$hasil = !\$a;
Penugasan	=	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> di dalam folder <code>dasarWeb</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 \$sisBagi = \$a % \$b; 10 \$pangkat = \$a ** \$b; </pre>
3	<p>Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/operator.php</code> Tampilkan di bawah ini (soal no 3.1) Catat di sini apa yang anda amati.</p> <p>Hasilnya:</p> <pre> Hasil Penjumlahan: 15 Hasil Pengurangan: 5 Hasil Perkalian: 50 Hasil Pembagian: 2 Sisa Pembagian: 0 Hasil Pangkat: 100000 </pre> <p>Penjelasan: Kode ini melakukan beberapa operasi matematika dasar menggunakan 2 variabel \$a dan \$b, kemudian dengan kode “echo” akan mencetak hasil dari operasi-operasi tersebut.</p>
4	Ketikkan kode tambahan pada langkah 5 di dalam <code>operator.php</code>
5	<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	<p>Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh localhost/dasarWeb/operator.php Tampilkan di bawah ini (soal no 3.2) Catat di sini apa yang anda amati.</p> <p>Hasilnya: Hasil Pengecekan Kesamaan: 0 Hasil Pengecekan Tidak Sama: 1 Hasil Pengecekan Lebih Kecil: 0 Hasil Pengecekan Lebih Besar: 1 Hasil Pengecekan Lebih Kecil atau Sama: 0 Hasil Pengecekan Lebih Besar atau Sama: 1</p> <p>Penjelasan: Kode ini melakukan beberapa operasi perbandingan antara nilai variabel \$a dan \$b, kemudian mengonversi hasilnya menjadi tipe data integer menggunakan (int). Kemudian, kode untuk mencetak hasil dari operasi-operasi perbandingan tersebut menggunakan "echo". Dimana nilai 1 menunjukkan bahwa pernyataan benar dan nilai 0 menunjukkan bahwa pernyataan salah.</p>
7	Ketikkan kode tambahan pada langkah 8 di dalam operator.php
8	<pre>19 \$hasilAnd = \$a && \$b; 20 \$hasilOr = \$a \$b; 21 \$hasilNotA = !\$a; 22 \$hasilNotB = !\$b;</pre>
9	<p>Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh localhost/dasarWeb/operator.php .Tampilkan di bawah ini (soal no 3.3) Catat di sini apa yang anda amati</p> <p>Hasilnya: Hasil Logika AND: 1 Hasil Logika OR: 1 Hasil Logika NOT untuk A: 0 Hasil Logika NOT untuk B: 0</p> <p>Penjelasan: Dalam kode ini, dilakukan beberapa operasi logika antara variabel \$a dan \$b, kemudian hasilnya dikonversi menjadi tipe data integer menggunakan (int). Maka akan muncul tampilan seperti pada gambar di atas.</p>
10	Ketikkan kode tambahan pada langkah 11 di dalam operator.php
11	<pre>24 \$a += \$b; 25 \$a -= \$b; 26 \$a *= \$b; 27 \$a /= \$b; 28 \$a %= \$b;</pre>
12	<p>Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh localhost/dasarWeb/operator.php Tampilkan di bawah ini (soal no 3.4) Catat di sini apa yang anda amati</p> <p>Hasilnya:</p>

	<p>Hasil Penambahan: 5 Hasil Pengurangan: 0 Hasil Perkalian: 0 Hasil Pembagian: 0 Sisa Pembagian: 0</p> <p>Penjelasan: Setiap operasi penugasan mengubah nilai variabel yang bersangkutan. Oleh karena itu, hasil dari masing-masing operasi akan berasal dari hasil operasi terakhir, bukan hasil setiap operasi yang ditambahkan dan dikurangkan secara berurutan.</p>
13	Ketikkan kode tambahan pada langkah 14 di dalam operator.php
14	<pre>30 \$hasilIdentik = \$a === \$b; 31 \$hasilTidakIdentik = \$a !== \$b;</pre>
15	<p>Lengkapi kode program di atas sehingga bisa menampilkan hasilnya dan rapi. Simpan file tersebut, kemudian buka browser dan jalankan/refresh localhost/dasarWeb/operator.php. Tampilkan di bawah ini (soal no 3.5) Catat di sini apa yang anda amati</p> <p>Hasilnya: Hasil Pengecekan Identik: false Hasil Pengecekan Tidak Identik: true</p> <p>Penjelasan: Kode tersebut melakukan operasi perbandingan identitas antara variabel \$a dan \$b menggunakan operator identitas (=== dan !==), dan kemudian mencetak hasilnya.</p>
16	Ada soal cerita : 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	<p>Buat kode program untuk langkah 16 dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 3.6)</p> <p>Kode Programnya:</p> <pre><?php // Jumlah total kursi di restoran \$totalKursi = 45; // Jumlah kursi yang telah ditempati oleh pelanggan \$kursiDitempati = 28; // Menghitung jumlah kursi yang masih kosong \$kursiKosong = \$totalKursi - \$kursiDitempati; // Menghitung persentase kursi yang masih kosong \$persenKursiKosong = (\$kursiKosong / \$totalKursi) * 100; // Menampilkan hasil echo "Jumlah kursi kosong: \$kursiKosong dari \$totalKursi kursi
"; echo "Persentase kursi kosong: \$persenKursiKosong%"; ?></pre>

Hasil Run di Browser:

Jumlah kursi kosong: 17 dari 45 kursi
Persentase kursi kosong: 37.777777777778%

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-langkah Praktikum :

Langkah	Keterangan
---------	------------

1	Buat satu file baru bernama struktur_kontrol.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam struktur_kontrol.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	<p>Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php</p> <p>Hasilnya:</p> <p>Nilai huruf: A</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <p>Penjelasan:</p> <p>Kode ini mengevaluasi nilai numerik yang disimpan dalam variabel \$nilaiNumerik dan mencetak nilai huruf yang sesuai dengan kriteria yang ditentukan. Seperti pada gambar di atas dimana yang tampil adalah huruf A yang berarti bahwa nilai tersebut berada diantara 90 dan 100.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam struktur_kontrol.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	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php</p> <p>Hasilnya:</p> <p>Atlet tersebut memerlukan 17 hari untuk mencapai jarak 500 kilometer.</p>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <p>Penjelasan:</p> <p>While akan terus berjalan selama jarak saat ini (\$jarakSaatIni) masih kurang dari jarak target (\$jarakTarget).</p>
9	Ketikkan kode tambahan pada langkah 9 di dalam struktur_kontrol.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	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php</p> <p>Hasilnya: Jumlah buah yang dipanen adalah: 500</p>
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.3)</p> <p>Penjelasan: Kode ini menggunakan perulangan for untuk menghitung total jumlah buah yang dipanen dari sejumlah lahan yang telah ditentukan. For akan terus berjalan sebanyak jumlah lahan yang dimiliki.</p>
13	<p>Ketikkan kode tambahan pada langkah 14 di dalam struktur_kontrol.php</p>
14	<pre> \$skorUjian = [85, 92, 78, 96, 88]; \$totalSkor = 0; foreach (\$skorUjian as \$skor) { \$totalSkor += \$skor; } echo "Total skor ujian adalah: \$totalSkor"; </pre>
15	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php</p> <p>Hasilnya: Total skor ujian adalah: 439</p>
16	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.4)</p> <p>Penjelasan: kode ini menggunakan loop foreach untuk menjumlahkan total skor dari serangkaian skor ujian yang disimpan dalam array \$skorUjian.</p>
17	<p>Ketikkan kode tambahan pada langkah 18 di dalam struktur_kontrol.php</p>

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	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php</p> <p>Hasilnya:</p> <pre> Nilai: 85 (Lulus) Nilai: 92 (Lulus) Nilai: 58 (Tidak Lulus) Nilai: 64 (Lulus) Nilai: 90 (Lulus) Nilai: 55 (Tidak Lulus) Nilai: 88 (Lulus) Nilai: 79 (Lulus) Nilai: 70 (Lulus) Nilai: 96 (Lulus) </pre>
20	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.5)</p> <p>Penjelasan:</p> <p>Kode ini menggunakan loop foreach ntukmengiterasi melalui setiap elemen dalam array \$nilaiSiswa, yang berisi nilai-nilai siswa. Kemudian, kode mengevaluasi setiap nilai dan mencetak apakah siswa tersebut lulus atau tidak.</p>
21	<p>Ada soal cerita : 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	<p>Buat kode program untuk langkah 21 dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 4.6)</p> <p>Kode Programnya:</p> <pre> <?php \$nilaiSiswa = [85, 92, 78, 64, 90, 75, 88, 79, 70, 96]; // Mengurutkan nilai siswa secara menaik sort(\$nilaiSiswa); // Mengabaikan dua nilai terendah dan dua nilai tertinggi \$nilaiTertinggiDihapus = array_slice(\$nilaiSiswa, 2, -2); // Menghitung total nilai yang tersisa \$totalNilai = array_sum(\$nilaiTertinggiDihapus); // Mencetak total nilai yang dihitung </pre>

	<pre>echo "Total nilai siswa setelah mengabaikan dua nilai tertinggi dan dua nilai terendah: \$totalNilai"; ?></pre> <p>Hasil Run:</p> <pre>Total nilai siswa setelah mengabaikan dua nilai tertinggi dan dua nilai terendah: 495</pre>
23	<p>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.</p>
24	<p>Buat kode program untuk langkah 23 dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 4.7)</p> <p>Kode Programnya:</p> <pre><?php // Harga produk \$hargaProduk = 120000; // Batas pembelian untuk mendapatkan diskon \$batasDiskon = 100000; // Persentase diskon \$persentaseDiskon = 20; // 20% // Menghitung harga setelah diskon if (\$hargaProduk > \$batasDiskon) { // Jika total pembelian melebihi batas diskon, terapkan diskon \$diskon = \$hargaProduk * (\$persentaseDiskon / 100); // Hitung jumlah diskon \$hargaSetelahDiskon = \$hargaProduk - \$diskon; // Kurangi harga produk dengan diskon } else { // Jika total pembelian tidak memenuhi syarat diskon, harga setelah diskon tetap sama dengan harga produk \$hargaSetelahDiskon = \$hargaProduk; } // Cetak harga yang harus dibayar setelah mendapatkan diskon echo "Harga yang harus dibayar setelah mendapatkan diskon: Rp " . number_format(\$hargaSetelahDiskon, 0, ',', '.'); ?></pre> <p>Hasil Run:</p> <pre>Harga yang harus dibayar setelah mendapatkan diskon: Rp 96.000</pre>
25	<p>Ada soal cerita : 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)”</p>

26	<p>Buat kode program untuk langkah 25 dengan adanya script Ternary dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 4.8)</p> <p>Kode Programnya:</p> <pre><?php // Skor pemain \$skorPemain = 700; // Total skor pemain \$totalSkorPemain = "Total skor pemain adalah: \$skorPemain"; // Apakah pemain mendapatkan hadiah tambahan? \$hadiahTambahan = (\$skorPemain > 500) ? "YA" : "TIDAK"; // Cetak total skor pemain dan apakah pemain mendapatkan hadiah tambahan menggunakan operator ternary echo \$totalSkorPemain . "
"; echo "Apakah pemain mendapatkan hadiah tambahan? " . \$hadiahTambahan; ?></pre> <p>Hasil Run:</p> <pre>Total skor pemain adalah: 700 Apakah pemain mendapatkan hadiah tambahan? YA</pre>
----	--

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"];
```

Anda dapat mengakses nilai dalam array dengan menggunakan 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
```

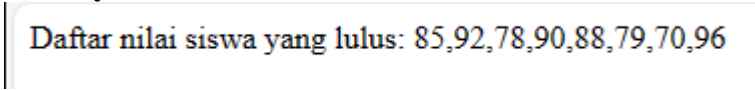
4. Iterasi Melalui Array Asosiatif

```
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-langkah Praktikum :

Langka h	Keterangan
1	Buat satu file baru bernama array.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam array.php
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 localhost/dasarWeb/array.php Hasilnya: 
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1) Penjelasan: Program tersebut memiliki tujuan untuk menyaring nilai-nilai siswa yang memenuhi syarat untuk lulus, yaitu nilai yang lebih besar atau sama dengan 70. Disini menggunakan perulangan foreach melalui setiap elemen dalam array \$nilaiSiswa.
5	Ketikkan kode tambahan pada langkah 6 di dalam array.php

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	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/array.php</p> <p>Hasilnya: Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: Alice, Charlie, Eva</p>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)</p> <p>Penjelasan: Program tersebut memiliki tujuan untuk menyaring karyawan yang memiliki pengalaman kerja lebih dari 5 tahun dari sebuah daftar karyawan yang tersedia. Menggunakan perulangan foreach melalui setiap elemen dalam array \$daftarKaryawan.</p>
9	<p>Ketikkan kode tambahan pada langkah 10 di dalam array.php</p>
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>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/array.php</p> <p>Hasilnya:</p> <p>Daftar nilai mahasiswa dalam mata kuliah Fisika: Nama: Alice, Nilai: 90 Nama: Bob, Nilai: 88 Nama: Charlie, Nilai: 75</p>
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p> <p>Penjelasan:</p> <p>Kode program ini bertujuan untuk mencetak daftar nilai mahasiswa dalam suatu mata kuliah tertentu dari sebuah array multidimensi yang menyimpan nilai-nilai mahasiswa dalam berbagai mata kuliah.</p>
13	<p>Ada soal cerita : 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 program untuk langkah 13 dengan array dua dimensi dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 5.4)</p> <p>Kode Programnya:</p> <pre><?php // Data nilai siswa \$dataNilai = ['Alice' => 85, 'Bob' => 92, 'Charlie' => 78, 'David' => 64, 'Eva' => 90,]; // Menghitung rata-rata nilai kelas \$rataRataKelas = array_sum(\$dataNilai) / count(\$dataNilai); // Mencetak daftar nilai siswa yang di atas rata-rata kelas echo "Daftar nilai siswa yang mencapai nilai di atas rata-rata kelas (\$rataRataKelas):
"; foreach (\$dataNilai as \$nama => \$nilai) { if (\$nilai > \$rataRataKelas) { echo "Nama: \$nama, Nilai: \$nilai
"; } } ?></pre> <p>Hasil Run:</p> <p>Daftar nilai siswa yang mencapai nilai di atas rata-rata kelas (81.8): Nama: Alice, Nilai: 85 Nama: Bob, Nilai: 92 Nama: Eva, Nilai: 90</p>

TUGAS

```
<?php
$biodata=[
    'Lentina'=>[
        [' ', ''],
        ['Nama      :', 'Lentina Abrila Putri'],
        ['Alamat    :', 'Jl. Ahmad Yani, Malang'],
        ['No. Tlp    :', '081517946770'],
        ['Jurusan    :', 'Teknologi Informasi'],
    ],
    'Maty'=>[
        [' ', ''],
        ['Nama      :', 'Matty Braps'],
        ['Alamat    :', 'Jl. Salak, Malang'],
        ['No. Tlp    :', '0819688222'],
        ['Jurusan    :', 'Teknik Kimia'],
    ]
];
// pada kode diatas data biodata orang diisi dalam array 2 dimensi

foreach ($biodata as $nama => $data) {
    echo "<table>";
    foreach ($data as $baris) {
        echo "<tr>";
        foreach ($baris as $isi) {
            echo "<td>$isi</td>";
        }
        echo "</tr>";
        echo "<br>";
    }
    echo "</table>";
}
// kode diatas untuk membuat tabel agar output menjadi lebih rapi
?>
```

Hasil Run:



Nama : Lentina Abrila Putri
Alamat : Jl. Ahmad Yani, Malang
No. Tlp : 081517946770
Jurusan : Teknologi Informasi



Nama : Matty Braps
Alamat : Jl. Salak, Malang
No. Tlp : 0819688222
Jurusan : Teknik Kimia