

# **PRAKTIKUM PEMOGRAMAN WEB**

## **Jobsheet 4: PHP**

**Dosen Pengajar: Dimas Wahyu Wibowo, S.T., M.T.**



**Oleh :**

Fardiyani Afro'ul Yasinta (2241760030)

Kelas SIB-2E /07

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**



Jurusan Teknologi Informasi Politeknik Negeri Malang

## **Jobsheet-4: PHP**

### **Mata Kuliah Desain dan Pemrograman Web**

Pengampu: Tim Ajar Desain dan Pemrograman Web

September 2023

---

#### **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"**;
- **Type 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
$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 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 &lt;?php 2 \$angka1 = 10; 3 \$angka2 = 5; 4 \$hasil = \$angka1 + \$angka2; 5 echo "Hasil penjumlahan \$angka1 dan \$angka2 adalah \$hasil."; 6 ?&gt;</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 ?&gt;</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 16 didirikan pada tahun " . TAHUN_PENDIRIAN . ".";</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><b>Hasil:</b></p> <p>Hasil penjumlahan 10 dan 5 adalah .Variabel benar: 1, Variabel salah: Selamat datang di WebsiteKu.com, situs yang didirikan pada tahun2023.</p>
9	<p>Apa yang anda pahami dari penggunaan variable pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1)</p> <p><b>Hasil:</b></p> <p>Hasil penjumlahan 10 dan 5 adalah 15.  Variabel benar: 1  Variabel salah:  Selamat datang di Websiteku.com.  Didirikan pada tahun 2023.</p> <p><b>Jawab:</b>  Pada percobaan sebelumnya tidak ada penggunaan &lt;br&gt; hal itu yang menjadikan kalimat hanya satu baris saja, setelah ditambahkan &lt;br&gt; baris kalimat sudah tidak menjadi satu baris.</p>

## Type 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  &lt;?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} &lt;br&gt;"; 9  echo "Variabel b: {\$b} &lt;br&gt;"; 10 echo "Variabel c: {\$c} &lt;br&gt;"; 11 echo "Variabel d: {\$d} &lt;br&gt;"; 12 echo "Variabel e: {\$e} &lt;br&gt;"; 13 14 var_dump(\$e); 15 ?&gt;</pre>
3	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p><b>Hasil:</b></p> <pre>Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 40 int(40)</pre>
4	Tambahkan isi dari file tipe_data.php tersebut dengan kode di bawah ini.

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} &lt;br&gt;"; 23  echo "IPA: {\$nilaiIPA} &lt;br&gt;"; 24  echo "Bahasa Indonesia: {\$nilaiBahasaIndonesia} &lt;br&gt;"; 25  echo "Rata-rata: {\$rataRata} &lt;br&gt;"; 26 27  var_dump(\$rataRata); 28  ?&gt; </pre>
6	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p><b>Hasil:</b></p> <p>Matematika: 5.1  IPA: 6.7  Bahasa Indonesia: 9.3  Rata-rata: 7.03333333333333</p>
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 "&lt;br&gt;"; 34  var_dump(\$apakahSiswaSudahUjian); 35  ?&gt; </pre>
9	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p><b>Hasil:</b></p> <p>bool(true)  bool(false)</p>
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} &lt;br&gt;"; 43  echo 'Nama Belakang: ' . \$namaBelakang . '&lt;br&gt;'; 44 45  echo \$namaLengkap; 46  ?&gt; </pre>
12	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php</p> <p><b>Hasil:</b></p> <p>Nama Depan: Ibnu  Nama Belakang: Jakaria  Ibnu Jakaria</p>
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  ?&gt; </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><b>Hasil:</b></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><b>Jawab:</b></p> <p>tanda kurung kurawal ({}) digunakan dalam konsep yang disebut sebagai "string interpolation" atau "heredoc" syntax. Dengan menggunakan tanda kurung kurawal, dapat menyisipkan nilai variabel ke dalam string tanpa harus menggunakan operator penggabung (titik). Sedangkan, ada juga yang tidak menggunakan kurung kurawa pada nama lengkap, penggunaan tanda kurung kurawal lebih sering digunakan untuk memudahkan pembacaan dan penulisan kode. Itu adalah salah satu fitur kenyamanan sintaksis dalam PHP.</p>

## 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: <b>\$hasil = \$angka1 + \$angka2;</b>
	–	Pengurangan: <b>\$hasil = \$angka1 – \$angka2;</b>
	*	Perkalian: <b>\$hasil = \$angka1 * \$angka2;</b>
	/	Pembagian: <b>\$hasil = \$angka1 / \$angka2;</b>
	%	Sisa Bagi: <b>\$sis = \$angka1 % \$angka2;</b>
Pembanding	==	Sama dengan: <b>\$hasil = \$nilai1 == \$nilai2;</b>
	!=	Tidak sama dengan: <b>\$hasil = \$nilai1 != \$nilai2;</b>
	<	Lebih kecil dari: <b>\$hasil = \$nilai1 &lt; \$nilai2;</b>
	>	Lebih besar dari: <b>\$hasil = \$nilai1 &gt; \$nilai2;</b>
	<=	Lebih kecil dari atau sama dengan: <b>\$hasil = \$nilai1 &lt;= \$nilai2;</b>
	>=	Lebih besar dari atau sama dengan: <b>\$hasil = \$nilai1 &gt;= \$nilai2;</b>
Logika	&& / and	AND (dan): <b>\$hasil = \$a &amp;&amp; \$b;</b>
	/ or	OR (atau): <b>\$hasil = \$a    \$b;</b>
	! / not	NOT (negasi): <b>\$hasil = !\$a;</b>

Penugasan	=	Penugasan biasa: <b><math>\\$a = \\$b;</math></b>
	+=	Penugasan dengan penjumlahan: <b><math>\\$a += \\$b;</math></b>
	-=	Penugasan dengan pengurangan: <b><math>\\$a -= \\$b;</math></b>
	*=	Penugasan dengan perkalian: <b><math>\\$a *= \\$b;</math></b>
	/=	Penugasan dengan pembagian: <b><math>\\$a /= \\$b;</math></b>
	%=	Penugasan dengan sisa bagi: <b><math>\\$a \% = \\$b;</math></b>
Increment/Decrement	++	Increment (menambah 1): <b><math>\\$a++;</math></b>
	—	Decrement (mengurangi 1): <b><math>\\$a--;</math></b>
Bitwise	&	AND bitwise: <b><math>\\$hasil = \\$a \&amp; \\$b;</math></b>
		OR bitwise: <b><math>\\$hasil = \\$a   \\$b;</math></b>
	^	XOR bitwise: <b><math>\\$hasil = \\$a \wedge \\$b;</math></b>
	~	NOT bitwise: <b><math>\\$hasil = \sim \\$a;</math></b>
Pemangkatan	**	Pangkat: <b><math>\\$hasil = \\$angka ** \\$pangkat;</math></b>
Identik	===	Identik (nilai dan tipe data sama): <b><math>\\$hasil = \\$a === \\$b;</math></b>
	!==	Tidak Identik (nilai atau tipe data tidak sama): <b><math>\\$hasil = \\$a !== \\$b;</math></b>
Ternary	? :	Operator kondisional (ternary): <b><math>\\$hasil = (\\$nilai &gt; 10) ? \text{“Lebih besar”} : \text{“Lebih kecil”};</math></b>
Array	[]	Akses elemen array: <b><math>\\$nilai = \\$array[0];</math></b>
	+	Penggabungan array: <b><math>\\$array3 = \\$array1 + \\$array2;</math></b>

### 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  &lt;?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  \$sisabagi = \$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><b>Hasil:</b></p> <hr/> <pre> Nilai a: 10 Nilai b: 5 Hasil Penjumlahan: 15 Hasil Pengurangan: 5 Hasil Perkalian: 50 Hasil Pembagian: 2 Sisa Bagi: 0 Hasil Pangkat: 100000</pre> <p>Kode di atas mendefinisikan dua variabel, <code>\$a</code> dan <code>\$b</code>, kemudian melakukan operasi aritmatika penjumlahan, pengurangan, perkalian, pembagian, sisa bagi, dan pangkat antara <code>\$a</code> dan <code>\$b</code>. Hasil-hasil operasi tersebut ditampilkan dalam sebuah pesan menggunakan fungsi <code>echo</code>.</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 &lt; \$b; 15 \$hasilLebihBesar = \$a &gt; \$b; 16 \$hasilLebihKecilSama = \$a &lt;= \$b; 17 \$hasilLebihBesarSama = \$a &gt;= \$b;</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><b>Hasil:</b></p> <pre> Hasil a sama dengan b: Hasil a tidak sama dengan b: 1 Hasil a lebih kecil dari b: Hasil a lebih besar dari b: 1 Hasil a lebih kecil dari atau sama dengan b: Hasil a lebih besar dari atau sama dengan b: 1 </pre> <p>Kode di atas menggunakan operator perbandingan untuk membandingkan nilai antara variabel \$a dan \$b. Hasil perbandingan tersebut disimpan dalam variabel-variabel baru, dan kemudian ditampilkan dalam pesan yang menjelaskan apakah pernyataan perbandingan tersebut benar atau salah menggunakan operator ternary</p>
7	Ketikkan kode tambahan pada langkah 8 di dalam operator.php
8	<pre> 19  \$hasilAnd = \$a &amp;&amp; \$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><b>Hasil:</b></p> <pre> Hasil dari operasi AND: 1 Hasil dari operasi OR: 1 Hasil dari operasi NOT untuk a: Hasil dari operasi NOT untuk b: </pre> <p>Kode tersebut menggunakan operator logika, seperti AND ('&amp;&amp;') berguna untuk mengembalikan nilai true jika kondisi keduanya benar, OR ('  ') berguna untuk mengembalikan nilai true jika salah satu kondisi bernilai benar, dan NOT ('!') jika nilai bernilai true not akan membuat false, untuk melakukan operasi logika antara variabel \$a dan \$b. Hasil operasi logika tersebut disimpan dalam variabel-variabel baru, dan kemudian ditampilkan dalam pesan yang menyatakan apakah pernyataan logika tersebut benar atau salah menggunakan operator ternary</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><b>Hasil:</b></p> <p>Nilai a setelah ditambah dengan b: 15          Nilai a setelah dikurangi dengan b: 10          Nilai a setelah dikalikan dengan b: 50          Nilai a setelah dibagi dengan b: 10          Nilai a setelah dimodulus dengan b: 0</p> <p>Pada kode di atas, operator penugasan seperti `+=`, `-=`, `*=`, `/=`, dan `%=` digunakan untuk memodifikasi nilai variabel `\$a` berdasarkan operasi aritmatika dengan nilai variabel `\$b`, dan kemudian hasilnya ditampilkan menggunakan `echo`.</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><b>Hasil:</b></p> <p>Hasil a identik dengan b:          Hasil a tidak identik dengan b: 1</p> <p>Pada kode di atas, operator identitas (`===` dan `!==`) digunakan untuk membandingkan nilai dan tipe data variabel `\$a` dan `\$b`. Hasil perbandingan tersebut disimpan dalam variabel variabel baru, dan kemudian ditampilkan dalam pesan yang menjelaskan apakah pernyataan identitas tersebut benar atau salah menggunakan operator ternary</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><b>Hasil:</b></p> <pre> Jumlah kursi di restoran: 45 Jumlah kursi yang telah ditempati: 28 Jumlah kursi yang masih kosong di restoran: 17 Persentase kursi yang masih kosong: 37.78% </pre>

### 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

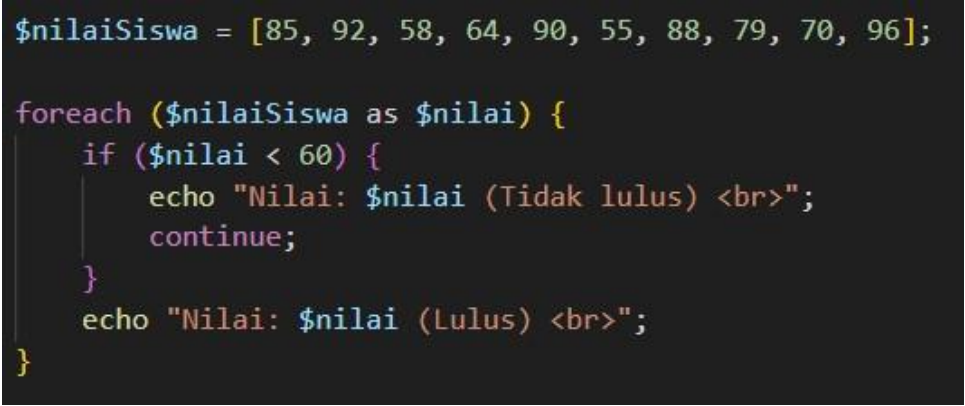
Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama <code>struktur_kontrol.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>struktur_kontrol.php</code>
2	<pre> 1  &lt;?php 2  \$nilaiNumerik = 92; 3 4  if (\$nilaiNumerik &gt;= 90 &amp;&amp; \$nilaiNumerik &lt;= 100) { 5      echo "Nilai huruf: A"; 6  } elseif (\$nilaiNumerik &gt;= 80 &amp;&amp; \$nilaiNumerik &lt; 90) { 7      echo "Nilai huruf: B"; 8  } elseif (\$nilaiNumerik &gt;= 70 &amp;&amp; \$nilaiNumerik &lt; 80) { 9      echo "Nilai huruf: C"; 10 } elseif (\$nilaiNumerik &lt; 70) { 11     echo "Nilai huruf: D"; 12 } 13 ?&gt;</pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/struktur_kontrol.php</code>

4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <p><b>Hasil:</b></p> <p>Nilai huruf: A</p> <p>Kode di atas merupakan implementasi dari struktur kondisional (`if-elseif-else`) untuk mengonversi nilai numerik ke dalam nilai huruf berdasarkan kriteria tertentu. Dalam hal ini, kriteria yang digunakan untuk menentukan nilai huruf adalah sebagai berikut:</p> <ul style="list-style-type: none"> <li>- Jika nilai numerik berada dalam rentang 90 hingga 100, maka nilai huruf adalah A.</li> <li>- Jika nilai numerik berada dalam rentang 80 hingga kurang dari 90, maka nilai huruf adalah B.</li> <li>- Jika nilai numerik berada dalam rentang 70 hingga kurang dari 80, maka nilai huruf adalah C.</li> <li>- Jika nilai numerik kurang dari 70, maka nilai huruf adalah D.</li> </ul>
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>struktur_kontrol.php</code>

6	<pre> \$jarakSaatIni = 0; \$jarakTarget = 500; \$peningkatanHarian = 30; \$hari = 0;  while (\$jarakSaatIni &lt; \$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 <code>localhost/dasarWeb/struktur_kontrol.php</code>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <p><b>Hasil:</b></p> <p>Atlet tersebut memerlukan 17 hari untuk mencapai jarak 500 kilometer.</p> <p>Kode di atas menggunakan loop `while` untuk menghitung berapa hari atlet tersebut memerlukan waktu untuk mencapai jarak 500 kilometer. Atlet tersebut setiap harinya menambahkan jarak sebesar 30 kilometer (`\$peningkatanHarian`) hingga mencapai atau melebihi jarak target sebesar 500 kilometer (`\$jarakTarget`).</p> <p>Hasil keluaran dari program ini adalah jumlah hari yang diperlukan atlet untuk mencapai atau melebihi jarak 500 kilometer.</p>
9	Ketikkan kode tambahan pada langkah 9 di dalam <code>struktur_kontrol.php</code>

10	<pre> \$jumlahLahan = 10; \$tanamanPerLahan = 5; \$buahPerTanaman = 10; \$jumlahBuah = 0;  for (\$i = 1; \$i &lt;= \$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 localhost/dasarWeb/struktur_kontrol.php
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.3)</p> <p><b>Hasil:</b></p> <pre> Jumlah buah yang akan dipanen adalah: 500 </pre> <p>Kode tersebut menggunakan loop `for` untuk menghitung jumlah buah yang akan dipanen berdasarkan beberapa variabel, yaitu ` \$jumlahLahan `: Jumlah lahan yang ditanami., ` \$tanamanPerLahan `: Jumlah tanaman per lahan., ` \$buahPerTanaman `: Jumlah buah yang dihasilkan oleh setiap tanaman., ` \$jumlahBuah `: Jumlah total buah yang akan dipanen. Dalam loop, setiap iterasi menambahkan jumlah buah sebanyak hasil perkalian dari jumlah tanaman per lahan dan jumlah buah per tanaman. Proses ini dilakukan sebanyak jumlah lahan yang ditanami. Hasil keluaran dari program ini adalah jumlah total buah yang akan dipanen.</p>
13	Ketikkan kode tambahan pada langkah 14 di dalam struktur_kontrol.php
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 localhost/dasarWeb/struktur_kontrol.php

16	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.4)</p> <p><b>Hasil:</b></p> <p>Total skor ujian adalah: 439</p> <p>Kode tersebut menggunakan loop `foreach` untuk menjumlahkan nilai-nilai dalam array `\$skorUjian`. Foreach sendiri digunakan untuk melakukan iterasi (perulangan) melalui setiap elemen dalam array atau objek pada PHP. Setiap nilai dalam array ditambahkan ke variabel `\$totalSkor`. Hasil keluaran dari program ini adalah total skor ujian, yang diperoleh dengan menjumlahkan semua nilai 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 &lt; 60) {         echo "Nilai: \$nilai (Tidak lulus) &lt;br&gt;";         continue;     }     echo "Nilai: \$nilai (Lulus) &lt;br&gt;"; } </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>
20	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.5)</p> <p><b>Hasil:</b></p> <p>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)</p> <p>Dalam kode di atas, digunakan array `\$nilaiSiswa` yang berisi nilai-nilai siswa dalam ujian. Penggunaan loop `foreach` membantu iterasi melalui setiap nilai dalam array dan menggunakan struktur kondisional `if` untuk menentukan apakah nilai tersebut lulus atau tidak lulus. Pada setiap iterasi, hasil keluaran menunjukkan status kelulusan siswa beserta nilai yang bersangkutan, dengan menggunakan pernyataan `continue` untuk melanjutkan iterasi ke nilai selanjutnya setelah menangani kasus nilai yang tidak lulus.</p>

21	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)
22	<p>Buat kode program untuk langkah 21 dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 4.6)</p> <p><b>Hasil:</b></p> <p>Nilai setelah dihilangkan satu nilai terendah dan satu nilai tertinggi: 75, 78, 79, 85, 88, 90  Total nilai setelah dihilangkan satu nilai terendah dan satu nilai tertinggi: 495  Rata-rata nilai setelah dihilangkan satu nilai terendah dan satu nilai tertinggi: 82.5</p>
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	<p>Buat kode program untuk langkah 23 dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 4.7)</p> <p><b>Hasil:</b></p> <p>Harga awal produk: Rp 120.000  Besaran diskon: Rp 24.000  Harga yang harus dibayar setelah mendapatkan diskon: Rp 96.000</p>
25	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)"
26	<p>Buat kode program untuk langkah 25 dengan adanya script <b>Ternary</b> dan tampilkan hasilnya di bawah ini beserta kode programnya (soal no 4.8)</p> <p><b>Hasil:</b></p> <p>Total skor pemain adalah: 551  Apakah pemain mendapatkan hadiah tambahan? (YA)</p>

## 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 &gt;= 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
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <p><b>Hasil:</b></p> <p>Daftar nilai siswa yang lulus: 85, 92, 78, 90, 88, 79, 70, 96</p> <p>Dalam kode di atas, sebuah array baru bernama ` \$nilaiLulus ` dibuat untuk menyimpan nilai siswa yang memenuhi kriteria kelulusan, yaitu nilai lebih besar atau sama dengan 70. Loop `foreach` kemudian digunakan untuk iterasi melalui setiap nilai dalam array ` \$nilaiSiswa `, dan nilai yang lulus ditambahkan ke dalam array ` \$nilaiLulus ` menggunakan ` \$nilaiLulus[] = \$nilai; `. Hasil dari program ini adalah daftar nilai siswa yang lulus, yang ditampilkan dengan menggunakan fungsi `implode` untuk menggabungkan nilai-nilai tersebut ke dalam sebuah string, dipisahkan oleh koma (`,`).</p>
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] &gt; 5) {         \$karyawanPengalamanLimaTahun[] = \$karyawan[0];     } }  echo "Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: " . implode(', ',     \$karyawanPengalamanLimaTahun); </pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/array.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)</p> <p><b>Hasil:</b></p> <p>Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: Alice, Charlie, Eva</p> <p>Dalam kode tersebut, terdapat array multidimensi ` \$daftarKaryawan ` yang menyimpan informasi nama karyawan dan tahun pengalaman kerja masing-masing karyawan. Sebuah array baru bernama ` \$karyawanPengalamanLimaTahun ` digunakan untuk menyimpan nama karyawan yang memiliki pengalaman kerja lebih dari 5 tahun. Loop `foreach` digunakan untuk iterasi melalui setiap elemen dalam array ` \$daftarKaryawan `. Pada setiap iterasi, dilakukan pengecekan apakah tahun pengalaman kerja karyawan lebih dari 5 tahun. Jika ya, maka nama karyawan tersebut ditambahkan ke dalam array ` \$karyawanPengalamanLimaTahun ` menggunakan ` \$karyawanPengalamanLimaTahun[] = \$karyawan[0]; `. Hasil keluaran dari program ini adalah daftar karyawan yang memiliki pengalaman kerja lebih dari 5 tahun, yang ditampilkan dengan menggunakan fungsi `implode` untuk menggabungkan nama-nama tersebut ke dalam sebuah string, dipisahkan oleh koma (`,`).</p>

9	Ketikkan kode tambahan pada langkah 10 di dalam array.php
10	<pre> \$daftarNilai = [     'Matematika' =&gt; [         ['Alice', 85],         ['Bob', 92],         ['Charlie', 78],     ],     'Fisika' =&gt; [         ['Alice', 90],         ['Bob', 88],         ['Charlie', 75],     ],     'Kimia' =&gt; [         ['Alice', 92],         ['Bob', 80],         ['Charlie', 85],     ], ];  \$mataKuliah = 'Fisika';  echo "Daftar nilai mahasiswa dalam mata kuliah \$mataKuliah: &lt;br&gt;";  foreach (\$daftarNilai[\$mataKuliah] as \$nilai) {     echo "Nama: {\$nilai[0]}, Nilai: {\$nilai[1]} &lt;br&gt;"; } </pre>
11	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/array.php
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p> <p><b>Hasil:</b></p> <p>Daftar nilai mahasiswa dalam mata kuliah Fisika:  Nama: Alice, Nilai: 90  Nama: Bob, Nilai: 88  Nama: Charlie, Nilai: 75</p> <p>Dalam kode di atas, terdapat array asosiatif multidimensi ` \$daftarNilai ` yang menyimpan nilai-nilai mahasiswa dalam beberapa mata kuliah. Variabel ` \$mataKuliah ` diinisialisasi dengan nama mata kuliah tertentu, dalam hal ini 'Fisika'. Loop `foreach` kemudian digunakan untuk iterasi melalui array nilai mahasiswa pada mata kuliah yang ditentukan. Setiap iterasi akan menampilkan nama dan nilai mahasiswa dalam mata kuliah tersebut menggunakan `echo`.</p> <p>Hasil keluaran dari program ini adalah daftar nilai mahasiswa dalam mata kuliah yang spesifik, dalam hal ini 'Fisika'.</p>
13	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

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><b>Hasil:</b></p> <p>Daftar nilai siswa yang di atas rata-rata kelas: Alice: 85 Bob: 92 Eva: 90</p>
----	---