

JOBSHEET 4
PHP
DESAIN DAN PEMROGRAMAN WEB



Disusun Oleh:
Muhammad Naufal Assyauqi Ardyawan
2241760046
Presensi 16 / Kelas 2E

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2024



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

| Langka h | 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 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/variabel_konstanta buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah |
| 9 | <p>Apa yang anda pahami dari penggunaan variable pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1)</p> <pre> Hasil Penjumlahan 10 dan 5 adalah 15 Variable benar: 1, Variable salah: Selamat Datang di Websiteku.com, situs yang didirikan pada tahun 2023. </pre> <p>Pada PHP, variable dapat di instansiasikan dengan perintah DEFINE, dan memiliki format "\$namaVariable". Variable yang telah di DEFINE dapat dipanggil dengan perintah ECHO untuk menampilkan isi nilai variable tersebut. Variable yang menyimpan nilai boolean (TRUE or FALSE) apabila dipanggil dengan echo, nilai TRUE akan menampilkan angka "1", dan nilai FALSE akan menampilkan nilai kosong.</p> |

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

| Langka h | 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 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php |
| 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} "; 23 echo "IPA: {\$nilaiIPA} "; 24 echo "Bahasa Indonesia: {\$nilaiBahasaIndonesia} "; 25 echo "Rata-rata: {\$rataRata} "; 26 27 var_dump(\$rataRata); 28 ?> </pre> |
| 6 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php |
| 7 | Tambahkan isi dari file tipe_data.php tersebut dengan kode di bawah ini. |
| 8 | <pre> 29 \$apakahSiswaLulus = true; 30 \$apakahSiswaSudahUjian = false; 31 32 var_dump(\$apakahSiswaLulus); 33 echo " "; 34 var_dump(\$apakahSiswaSudahUjian); 35 ?> </pre> |
| 9 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/tipe_data.php |

| | |
|----|--|
| 10 | Tambahkan isi dari file <code>tipe_data.php</code> tersebut dengan kode di bawah ini. |
| 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 | Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/tipe_data.php</code> |
| 13 | Tambahkan isi dari file <code>tipe_data.php</code> tersebut dengan kode di bawah ini. |
| 14 | <pre> 47 \$listMahasiswa = ["Wahid Abdullah", "Elmo Bachtiar", "Lendis Fabri"]; 48 echo \$listMahasiswa[0]; 49 ?> </pre> |
| 15 | Rapikan hasil supaya ada jarak per tipe data. Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/tipe_data.php</code> |
| 16 | <p>Apa yang anda pahami dari penggunaan tipe data pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2)</p> <pre> Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 70 int(70) Matematika: 5.1 IPA: 6.7 Bahasa Indonesia: 9.3 Rata - rata: 7.03333333333333 string(16) "7.03333333333333" " bool(true) bool(false) Nama Depan: Ibnu Nama Belakang : Jakaria Ibnu Jakaria Wahid Abdullah </pre> <p>Nilai berbentuk angka (int, float) yang disimpan pada sebuah variable dapat digunakan dalam operasi perhitungan. Nilai yang berbentuk teks (String) dapat digabung menjadi satu ke dalam variable baru dengan 2 cara: 1, dengan menggunakan tanda kutip 2 dalam sebuah baris ECHO</p> |

dan menempatkan variable yang ingin dimasukkan diantara simbol kurung kurawal, dan 2, tanpa menggunakan kutip dua dan langsung menempatkan variable dengan menggunakan simbol “.” untuk menambahkannya ke sebuah baris ECHO.

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: \$hasil = \$angka1 + \$angka2; |
| | – | Pengurangan: \$hasil = \$angka1 – \$angka2; |
| | * | Perkalian: \$hasil = \$angka1 * \$angka2; |
| | / | Pembagian: \$hasil = \$angka1 / \$angka2; |
| | % | Sisa Bagi: \$sisa = \$angka1 % \$angka2; |
| Pembanding | == | 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; |

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

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 \$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 localhost/dasarWeb/operator.php Tampilkan di bawah ini (soal no 3.1) Catat di sini apa yang anda amati</p> <p>Modifikasi:</p> <pre><?php \$a = 10; \$b = 5; \$hasilTambah = \$a + \$b; echo ("Hasil Tambah = {\$hasilTambah}\n"); \$hasilKurang = \$a - \$b; echo ("Hasil Kurang = {\$hasilKurang}\n"); \$hasilKali = \$a * \$b; echo ("Hasil Kali = {\$hasilKali}\n"); \$hasilBagi = \$a / \$b; echo ("Hasil Bagi = {\$hasilBagi}\n"); \$sisabagi = \$a % \$b; echo ("Sisa Bagi = {\$sisabagi}\n"); \$pangkat = \$a ** \$b; echo ("Hasil Pangkat = {\$pangkat} \n");</pre> <p>Output:</p> <pre>Hasil Tambah = 15 Hasil Kurang = 5 Hasil Kali = 50 Hasil Bagi = 2 Sisa Bagi = 0 Hasil Pangkat = 100000</pre> <p>Sebuah variable dapat menyimpan hasil operasi matematika dasar.</p> |
| 4 | Ketikkan kode tambahan pada langkah 5 di dalam operator.php |
| 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>Modifikasi:</p> |

| | |
|---|--|
| | <pre> \$hasilSama = \$a == \$b; echo ("Hasil Sama = {\$hasilSama} \n"); \$hasilTidakSama = \$a != \$b; echo ("Hasil Tidak Sama = {\$hasilTidakSama} \n"); \$hasilLebihKecil = \$a < \$b; echo ("Hasil Lebih Kecil = {\$hasilLebihKecil} \n"); \$hasilLebihBesar = \$a > \$b; echo ("Hasil Lebih Besar = {\$hasilLebihBesar} \n"); \$hasilLebihKecilSama = \$a <= \$b; echo ("Hasil Lebih Kecil Sama = {\$hasilLebihKecilSama} \n"); \$hasilLebihBesarSama = \$a >= \$b; echo ("Hasil Lebih Besar Sama= {\$hasilLebihBesarSama} \n"); </pre> <p>Output:</p> <pre> Hasil Sama = Hasil Tidak Sama = 1 Hasil Lebih Kecil = Hasil Lebih Besar = 1 Hasil Lebih Kecil Sama = Hasil Lebih Besar Sama= 1 </pre> <p>Sebuah variable juga dapat menyimpan nilai boolean yang didapatkan dari operator perbandingan antara dua nilai angka. angka 1 bernilai TRUE, dan nilai kosong bernilai FALSE.</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>Modifikasi:</p> <pre> \$hasilAnd = \$a && \$b; \$hasilOr = \$a \$b; \$hasilNotA = !\$a; \$hasilNotB = !\$b; echo("Hasil AND = {\$hasilAnd} \n"); echo("Hasil OR = {\$hasilOr} \n"); echo("Hasil NOT A = {\$hasilNotA} \n"); echo("Hasil NOT B = {\$hasilNotB} \n"); </pre> <p>Output:</p> |

| | |
|----|--|
| | <pre> Hasil AND = 1 Hasil OR = 1 Hasil NOT A = Hasil NOT B = </pre> <p>Variable juga dapat menyimpan nilai boolean dari operator logic gate.</p> |
| 10 | Ketikkan kode tambahan pada langkah 11 di dalam <code>operator.php</code> |
| 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 <code>localhost/dasarWeb/operator.php</code> Tampilkan di bawah ini (soal no 3.4) Catat di sini apa yang anda amati</p> <p>Modifikasi:</p> <pre> \$a += \$b; echo("Hasil A-1 = {\$a} \n"); \$a -= \$b; echo("Hasil A-2 = {\$a} \n"); \$a *= \$b; echo("Hasil A-3 = {\$a} \n"); \$a /= \$b; echo("Hasil A-4 = {\$a} \n"); \$a %= \$b; echo("Hasil A-5 = {\$a} \n"); </pre> <p>Output:</p> <pre> Hasil A-1 = 15 Hasil A-2 = 10 Hasil A-3 = 50 Hasil A-4 = 10 Hasil A-5 = 0 </pre> <p>Operator yang digunakan langsung kepada variable utama akan merubah nilai awal variable tersebut. Dalam kasus ini, variable \$a yang awalnya bernilai 10, setiap dilakukan operasi perhitungan, nilai nya akan berubah menjadi 15, 10, 50, 10, dan 0.</p> |

| | | | |
|---|--|---|---|
| 13 | Ketikkan kode tambahan pada langkah 14 di dalam <code>operator.php</code> | | |
| 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 <code>localhost/dasarWeb/operator.php</code> Tampilkan di bawah ini (soal no 3.5) Catat di sini apa yang anda amati</p> <table border="1"> <tr> <td> Modifikasi: <pre> \$hasilIdentik = \$a === \$b; \$hasilTidakIdentik = \$a !== \$b; echo("\n"); echo("Hasil Identik = {\$hasilIdentik} \n"); echo("Hasil Tidak Identik = {\$hasilTidakIdentik} \n"); </pre> </td><td> Output: <pre> Hasil Identik = Hasil Tidak Identik = 1 </pre> </td></tr> </table> <p>Operator “===” dan “!==” memberikan nilai boolean dengan membandingkan kemiripan (ada atau tidaknya satu karakter yang sama) antara dua nilai variable. Karena nilai \$a = 0 dan nilai \$b = 5, tidak ada karakter yang sama, alhasil output yang dihasilkan seperti itu.</p> | Modifikasi: <pre> \$hasilIdentik = \$a === \$b; \$hasilTidakIdentik = \$a !== \$b; echo("\n"); echo("Hasil Identik = {\$hasilIdentik} \n"); echo("Hasil Tidak Identik = {\$hasilTidakIdentik} \n"); </pre> | Output: <pre> Hasil Identik = Hasil Tidak Identik = 1 </pre> |
| Modifikasi: <pre> \$hasilIdentik = \$a === \$b; \$hasilTidakIdentik = \$a !== \$b; echo("\n"); echo("Hasil Identik = {\$hasilIdentik} \n"); echo("Hasil Tidak Identik = {\$hasilTidakIdentik} \n"); </pre> | Output: <pre> Hasil Identik = Hasil Tidak Identik = 1 </pre> | | |
| 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 Program:</p> <pre> <?php \$kursiTotal = 45; \$kursiPenuh = 28; \$kursiSisa = \$kursiTotal - \$kursiPenuh; \$kursiSisaPersen = \$kursiSisa / \$kursiTotal * 100; echo("Persentase Sisa Kursi adalah " . number_format((float)\$kursiSisaPersen,1 ,'.', '') . "% \n"); ?> </pre> <p>Output:</p> <pre> Persentase Sisa Kursi adalah 37.8% </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 :

| Langka h | 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 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php |
| 4 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <p>Nilai Huruf : A</p> <p>Program melakukan pengecekan nilai dengan sintaks IF ... ELSE Dalam kasus ini, program mengecek nilai dari variable \$nilaiNumerik dan akan menjalankan baris kode echo yang sesuai. Karena nilai variable 92 dan berada diantara 90 dan 100, maka kode yang dijalankan adalah echo "Nilai Huruf : A".</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 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/struktur_kontrol.php |
| 8 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <p>Atlet tersebut memerlukan 17 hari untuk mencapai jarak 500 kilometer.</p> <p>Program melakukan perulangan dengan sintaks WHILE. Dalam kasus ini, perulangan akan terus dilakukan selama nilai \$jarakSaatIni lebih kecil dari \$jarakTarget. Setiap kali perulangan dilakukan, program akan menambahkan nilai \$peningkatanHarian kepada variable \$jarakSaatIni, dan melakukan increment pada variable \$hari. Perulangan selesai setelah \$jarakSaatIni lebih besar dari \$jarakTarget dan membutuhkan 17 kali iterasi.</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 <= \$jumlahLahan; \$i++) { \$jumlahBuah += (\$tanamanPerLahan * \$buahPerTanaman); } echo "Jumlah buah yang akan dipanen adalah: \$jumlahBuah"; </pre> |
| 11 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/struktur_kontrol.php</code> |
| 12 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.3)</p> <p>Jumlah buah yang akan dipanen adalah 500</p> <p>Program melakukan perulangan dengan sintaks FOR. Perulangan dilakukan selama \$i tidak melebihi nilai \$jumlahLahan, dan setiap iterasi dilakukan, nilai \$i di-increment.</p> |
| 13 | Ketikkan kode tambahan pada langkah 14 di dalam <code>struktur_kontrol.php</code> |
| 14 | <pre> \$skorUjian = [85, 92, 78, 96, 88]; \$totalSkor = 0; foreach (\$skorUjian as \$skor) { \$totalSkor += \$skor; } echo "Total skor ujian adalah: \$totalSkor"; </pre> |
| 15 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/struktur_kontrol.php</code> |
| 16 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.4)</p> <p>Total skor ujian adalah : 439</p> <p>Program melakukan perulangan dengan sintaks FOREACH. Perulangan dilakukan sebanyak jumlah indeks pada sebuah array (dalam kasus ini, jumlah indeks array \$skorUjian adalah 5, maka perulangan dilakukan 5 kali). Tiap nilai pada array \$skorUjian di alias kan dengan \$skor yang akan ditambahkan pada \$totalSkor.</p> |
| 17 | Ketikkan kode tambahan pada langkah 18 di dalam <code>struktur_kontrol.php</code> |

| | |
|----|--|
| 18 | <pre> \$nilaiSiswa = [85, 92, 58, 64, 90, 55, 88, 79, 70, 96]; foreach (\$nilaiSiswa as \$nilai) { if (\$nilai < 60) { echo "Nilai: \$nilai (Tidak lulus) "; continue; } echo "Nilai: \$nilai (Lulus) "; } </pre> |
| 19 | <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> <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> <p>Program melakukan perulangan dengan sintaks FOREACH, kemudian dalam tiap iterasi melakukan pemilihan IF ELSE dengan menggunakan nilai variable \$nilai dari tiap indeks array \$nilaiSiswa. Apabila nilai dibawah 60, maka program menjalankan ECHO “Nilai (tidak lulus)” kemudian sintaks CONTINUE; akan langsung melanjutkan perulangan FOREACH. Apabila nilai tidak memenuhi kondisi IF, maka program akan menjalankan ECHO “Nilai (lulus)”. Perulangan dilakukan sebanyak indeks pada array \$nilaiSiswa.</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 Program:</p> <pre> <?php \$nilaiSiswa = [85, 92, 58, 64, 90, 55, 88, 79, 70, 96]; \$nilaiTertinggiPertama = 0; \$nilaiTertinggiKedua = 0; \$nilaiTerendahPertama = 100; \$nilaiTerendahKedua = 100; </pre> |

| | |
|----|--|
| | <pre> foreach (\$nilaiSiswa as \$nilai) { if (\$nilai > \$nilaiTertinggiPertama) { \$nilaiTertinggiKedua = \$nilaiTertinggiPertama; \$nilaiTertinggiPertama = \$nilai; } elseif (\$nilai > \$nilaiTertinggiKedua) { \$nilaiTertinggiKedua = \$nilai; } if (\$nilai < \$nilaiTerendahPertama) { \$nilaiTerendahKedua = \$nilaiTerendahPertama; \$nilaiTerendahPertama = \$nilai; } elseif (\$nilai < \$nilaiTerendahKedua) { \$nilaiTerendahKedua = \$nilai; } } \$totalNilai = array_sum(\$nilaiSiswa) - (\$nilaiTertinggiPertama + \$nilaiTertinggiKedua + \$nilaiTerendahPertama + \$nilaiTerendahKedua); \$jumlahNilai = count(\$nilaiSiswa) - 4; \$rataRata = \$totalNilai / \$jumlahNilai; echo "Nilai rata-rata setelah mengabaikan dua nilai tertinggi dan terendah adalah: " . round(\$rataRata, 2); echo "\n"; ?> </pre> <p>Output:</p> <pre> Nilai rata-rata setelah mengabaikan dua nilai tertinggi dan terendah adalah: 79.33 </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 Program:</p> <pre> <?php \$hargaProduk = 120000; \$diskonPersen = 20; if (\$hargaProduk > 100000) { \$hargaSetelahDiskon = \$hargaProduk - (\$hargaProduk * (\$diskonPersen / 100)); } else { \$hargaSetelahDiskon = \$hargaProduk; } </pre> |

| | |
|----|--|
| | <pre>echo "Harga setelah diskon: Rp {\$hargaSetelahDiskon}"; echo "\n"; ?></pre> <p>Output:</p> <pre>Harga setelah diskon: Rp 96000</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 Program:</p> <pre>Total skor pemain adalah: 2050 Apakah pemain mendapatkan hadiah tambahan? YA</pre> <p>Output:</p> <pre><?php \$skor = [300, 600, 450, 700]; \$totalSkor = 0; foreach (\$skor as \$nilai) { \$totalSkor += \$nilai; } echo "Total skor pemain adalah: \$totalSkor \n"; echo "Apakah pemain mendapatkan hadiah tambahan? "; \$hasil = (\$totalSkor > 500)? "YA" : "TIDAK"; echo \$hasil; echo "\n"; ?></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:

```
$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 |

| | |
|---|---|
| 4 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <p>Daftar nilai siswa yang lulus: 85,92,78,90,88,79,96</p> <p>Program menginisialisasi 2 array, \$nilaiSiswa yang berisi semua nilai, dan \$nilaiLulus yang merupakan array kosong. Program kemudian melakukan pengecekan dengan FOREACH dan IF. Apabila nilai lebih dari 70, maka nilai tersebut dimasukkan ke dalam array \$nilaiLulus. Setelah perulangan selesai, program menampilkan semua isi array \$nilaiLulus dengan sintaks IMPLODE.</p> |
| 5 | <p>Ketikkan kode tambahan pada langkah 6 di dalam array.php</p> |
| 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> |
| 8 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)</p> <p>Daftar nilai siswa yang lulus: 85,92,78,90,88,79,96</p> <p>Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: Alice,Charlie,Eva</p> <p>Program menginisialisasikan array \$daftarKaryawan yang tiap indeksnya menyimpan nilai String nama karyawan, dan nilai int tahun pengalaman. Program juga membuat array kosong \$karyawanPengalamanLimaTahun. Program melakukan pengecekan seperti praktikum sebelumnya, apabila nilai tahun pengalaman lebih besar dari 5, maka nilai nama karyawan dimasukkan ke array kosong tadi. Setelah perulangan selesai, program menampilkan isi array \$karyawanPengalamanLimaTahun dengan sintaks IMPLODE.</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> |
| 12 | <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p> <pre> Daftar nilai siswa yang lulus: 85,92,78,90,88,79,96 Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: Alice,Charlie,Eva Daftar nilai mahasiswa dalam mata kuliah Fisika: Nama: Alice, Nilai: 90 Nama: Bob, Nilai: 88 Nama: Charlie, Nilai: 75 </pre> <p>Program menginisialisasi array \$daftarNilai yang tiap indeksnya menyimpan sebuah array untuk tiap mata kuliah, dan tiap array mata kuliah menyimpan array nama dan nilai mahasiswa. Kemudian program menginisialisasikan variable \$mataKuliah yang bernilai String 'Fisika'. Program melakukan perulangan FOREACH dengan menggunakan array Fisika di dalam array \$daftarNilai. Kemudian program melakukan ECHO nama dan nilai dari mahasiswa pada array Fisika.</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> |

Kode Program:

```
<?php
$daftarNilai = [
    'Matematika' => [
        ['Alice', 85],
        ['Bob', 92],
        ['Charlie', 78],
        ['David', 64],
        ['Eva', 90]
    ],
];

$mataKuliah = 'Matematika';

$totalNilai = 0;
$jumlahSiswa = count($daftarNilai[$mataKuliah]);
foreach ($daftarNilai[$mataKuliah] as $siswa) {
    $totalNilai += $siswa[1];
}

$rataRataKelas = $totalNilai / $jumlahSiswa;

$nilaiLulus = [];
foreach ($daftarNilai[$mataKuliah] as $siswa) {
    if ($siswa[1] > $rataRataKelas) {
        $nilaiLulus[] = $siswa[0] . " (" . $siswa[1] . ")";
    }
}

echo "Daftar nilai siswa yang mencapai nilai di atas rata-rata:\n";
echo implode("\n", $nilaiLulus);
echo "\n";
?>
```

Output:

```
Daftar nilai siswa yang mencapai nilai di atas rata-rata:
Alice (85)
Bob (92)
Eva (90)
```