

MODUL

PEMOGRAMAN WEB II

Oleh:

CHALIFA CHAZAR

MODUL 6

Pemrosesan Array

Tujuan:

Mahasiswa memahami penggunaan serta pendefinisian array untuk menyelesaikan permasalahan tertentu dalam kode PHP.

Pustaka:

Raharjo, B. (2015): *Mudah Belajar PHP Teknik Penggunaan Fitur-Fitur Baru Dalam PHP 5*. Informatika. Bandung

Prasetio, A. (2015): *Buku Pintar Webmaster*.

www.w3schools.com

1. Array

Array adalah suatu pengenal di dalam program yang fungsinya menampung beberapa nilai (lebih dari satu) nilai yang saling berkaitan atau berhubungan. Nilai yang terdapat pada suatu array disebut elemen array.

Dalam bahasa pemograman lainnya (C, C++, dan Java), setiap elemen array harus memiliki tipe data yang sama. Dalam PHP, elemen array bisa memiliki tipe data yang berbeda-beda.

Contoh:

```
<?php
    $a = array("budi", 20, 58.5);
?>
```

2. Jenis Array

Dalam PHP, array diklasifikasikan menjadi dua jenis, yaitu:

- Array berindeks (indexed array)
- Array asosiatif (associative array)

2.1. Array Berindeks

Array berindeks adalah array yang diindeks berdasarkan nomor/angka. Indeks array pada umumnya dimulai dari angka 0. Anda bebas mendefinisikan indeks dengan nilai yang Anda tentukan.

10	20	30	40	50
\$a[0]	\$a[1]	\$a[2]	\$a[3]	\$a[4]

Contoh diatas menunjukkan array dengan 5 buah elemen. Elemen pertama (\$a[0]) bernilai 10, elemen kedua (\$a[1]) bernilai 20, dan seterusnya. Dalam array berindeks, antara kunci (indeks) dan nilai tidak memiliki keterkaitan.

2.2. Array Asosiatif

Array asosiatif adalah array yang diindeks berdasarkan nama tertentu. Letak perbedaan antara array berindeks dan array asosiatif adalah hanya terletak pada penamaan indeksnya saja.

10	20	30	40	50
\$a["satu"]	\$a["dua"]	\$a["tiga"]	\$a["empat"]	\$a["lima"]

Array \$a diindeks berdasarkan nama, bukan berdasarkan nomor. Pada contoh diatas indeks array bertipe string. Pada umumnya array asosiatif digunakan untuk merepresentasikan sesuatu yang kunci dan nilainya memiliki keterkaitan, misalnya sebagai berikut.

```
<?php
$kota = array("jkt" => "jakarta", "bdg" => "bandung", "sby" => "surabaya");
?>
```

Pada contoh diatas, kunci "jkt" memiliki kaitan dengan "jakarta". Pada array \$kota, kita mendefinisikan kunci (*key*) sebagai singkatan dari nilai (*value*).

3. Membuat Array

Array dapat dibuat melalui dua cara, yaitu dengan menggunakan fungsi `array()` atau dengan membuat elemen-elemen array dan mengisi nilai-nilai ke dalam elemen-elemen tersebut secara langsung.

Berikut ini contoh pembuatan array dengan menggunakan fungsi `array()`.

- Untuk array berindeks:

```
<?php
$matakuliah = array ("pemograman web", "database", "keamanan
jaringan", "sistem informasi", "rekayasa perangkat lunak");
?>
```

- Untuk array asosiatif:

```
<?php
$detailmk = array ("kode" => "TKB5218",
                  "nama" => "pemograman web 2",
                  "sks" => 2);
?>
```

Berikut ini contoh pembuatan array dengan cara langsung membuat variabel array dan mengisi nilai ke dalamnya.

- Untuk array berindeks:

```
<?php
$matakuliah[0] = "pemograman web";
$matakuliah[1] = "database";
$matakuliah[2] = "keamanan jaringan";
$matakuliah[3] = "sistem informasi";
$matakuliah[4] = "rekayasa perangkat lunak";
?>
```

- Untuk array asosiatif:

```
<?php
$detailmk["kode"] = "TKB5218";
$detailmk["nama"] = "pemograman web 2";
$detailmk["sks"] = 2;
?>
```

4. Mengakses Elemen Array

Setelah array dibuat, langkah selanjutnya adalah mengakses nilai-nilai yang terkandung di dalamnya. Cara mengakses elemen array sangatlah sederhana. Karena elemen array berupa nilai, maka kita dapat memperlakukannya seperti layaknya variabel.

Anda dapat menempatkan nilai yang diakses ke dalam suatu variabel atau dapat juga diproses secara langsung baik dalam perhitungan maupun ditampilkan langsung.

Berikut ini adalah contoh mengakses nilai array ke dalam suatu variabel.

```
<?php
//menampung elemen array dalam variabel - array berindeks
$kategori = $matakuliah[0];
echo $kategori;

//menampung elemen array dalam variabel - array asosiatif
$detail = $detailmk["nama"];
echo $detail;
?>
```

Berikut ini adalah contoh mengakses elemen array secara langsung

```
<?php
//membuat kategori array berindeks
$matakuliah[0] = "pemograman web";
$matakuliah[1] = "database";
$matakuliah[2] = "keamanan jaringan";
$matakuliah[3] = "sistem informasi";
$matakuliah[4] = "rekayasa perangkat lunak";

//membuat kategori array asosiatif
$detailmk = array("kode" => "TKB5218", "nama" => "pemograman web 2", "sks"
=> 2);

//mengakses array berindeks
echo "DAFTAR MATAKULIAH" . '<br>';
echo $matakuliah[0] . '<br>';
echo $matakuliah[1] . '<br>';
echo $matakuliah[2] . '<br>';
echo $matakuliah[3] . '<br>';
echo $matakuliah[4] . '<br>';

//mengakses array asosiatif
echo '<br>' . "DETAIL MATAKULIAH" . '<br>';
echo "KODE : " . $detailmk["kode"] . '<br>';
echo "NAMA : " . $detailmk["nama"] . '<br>';
echo "JUMLAH SKS : " . $detailmk["sks"] . '<br>';
?>
```

Untuk mempermudah dalam melakukan pengaksesan elemen-elemen array, kita dapat menggunakan struktur pengulangan seperti while, do-while, for, maupun foreach. Namun untuk array jenis asosiatif biasanya menggunakan pengulangan foreach.

Tugas.

Buat program seperti pada contoh sebelumnya untuk mengakses array dengan menggunakan pengulangan. Baik untuk contoh array berindeks maupun array asosiatif.

5. Mengubah Nilai Pada Elemen Array

Nilai array dapat diubah sesuai dengan kebutuhan program. Caranya sama seperti saat proses pengisian nilai ke dalam elemen array. Nilai elemen array yang lama akan ditimpa oleh nilai yang baru.

```
<?php
//membuat array
$a[0] = 10;
$a[1] = 20;

//menampilkan array
echo "Nilai array sebelum diubah : " . '<br>';
echo $a[0] . '<br>';
echo $a[1] . '<br>';

//merubah nilai array
$a[0] = 100;
$a[1] = 200;

//menampilkan array setelah diubah
echo "Nilai array setelah diubah : " . '<br>';
echo $a[0] . '<br>';
echo $a[1] . '<br>';
?>
```

6. Menambah dan Menghapus Elemen Array

Cara sederhana untuk menambahkan elemen baru ke dalam suatu array adalah dengan membuat pasangan indeks/kunci-kunci baru.

Perhatikan contoh penambahan elemen array pada array berindeks dibawah ini.

```
//membuat array
$kota = array("Jakarta", "Bandung", "Surabaya");

//menampilkan array
echo "Nilai array sebelum ditambah : " . '<br>';
foreach ($kota as $k) {
    echo $k . '<br>';
}

//menambahkan nilai array
$kota[3] = "Semarang";
$kota[4] = "Denpasar";

//menampilkan array setelah diubah
echo '<br>' . "Nilai array setelah ditambah : " . '<br>';
foreach ($kota as $k) {
    echo $k . '<br>';
}
?>
```

Perhatikan contoh penambahan elemen array pada array asosiatif dibawah ini.

```
<?php
//membuat array
$kota = array("Jakarta", "Bandung", "Surabaya");

//menampilkan array
echo "Nilai array sebelum ditambah : " . '<br>';
foreach ($kota as $k) {
    echo $k . '<br>';
}

//menambahkan nilai array
$kota[3] = "Semarang";
$kota[4] = "Denpasar";

//menampilkan array setelah diubah
echo '<br>' . "Nilai array setelah ditambah : " . '<br>';
foreach ($kota as $k) {
    echo $k . '<br>';
}
?>
```

Untuk menghapus elemen array yang terdapat pada indeks tertentu, kita dapat menggunakan fungsi `unset()`. Perhatikan contoh berikut ini.

```
<?php
//membuat array
$kota = array("Jakarta", "Bandung", "Surabaya", "Semarang", "Denpasar",
"Yogyakarta");

//menampilkan array
echo "Nilai array sebelum ditambah : " . '<br>';
foreach ($kota as $k) {
    echo $k . '<br>';
}

//menambahkan nilai array
unset($kota[1]);
unset($kota[3]);

//menampilkan array setelah diubah
echo '<br>' . "Nilai array setelah ditambah : " . '<br>';
foreach ($kota as $k) {
    echo $k . '<br>';
}
?>
```

7. Fungsi-Fungsi yang Berhubungan dengan Array

Berikut ini beberapa fungsi yang dapat digunakan pada variabel bertipe array.

Fungsi	Kegunaan & Bentuk Umum
<code>is_array()</code>	Untuk memeriksa apakah suatu variabel bertipe array atau bukan Bentuk Umum: <code>is_array(\$nilai)</code>
<code>in_array()</code>	Untuk mencari \$nilai di dalam \$array Bentuk Umum: <code>bool in_array(mixed \$nilai, array \$array [, bool \$strict=false])</code>
<code>array_keys()</code>	Fungsi ini akan mengembalikan daftar nilai kunci (berupa numerik maupun string) yang terdapat di dalam array Bentuk umum: <code>array array_keys(array \$array [,mixed \$nilai [,bool \$strict = false]])</code>
<code>array_key_exists()</code>	Fungsi ini digunakan untuk memeriksa apakah suatu kunci tertentu terdapat di dalam array atau tidak Bentuk umum: <code>bool array_key_exists(mixed \$key, array \$array)</code>
<code>array_values()</code>	Fungsi ini akan mengembalikan daftar nilai yang terdapat di dalam array. Bentuk umum: <code>array array_values(array \$array)</code>

Fungsi	Kegunaan & Bentuk Umum
array_search()	Fungsi ini mengembalikan indeks dari nilai yang ditemukan di dalam array Bentuk umum: mixed array_search(mixed \$nilai, array \$array [, bool \$strict = false])
count() dan sizeof()	Untuk mengetahui jumlah elemen di dalam array Bentuk umum: count(\$nilai) dan/atau sizeof(\$nilai)
sort()	Fungsi ini akan mengurutkan (<i>ascending</i>) array berdasarkan nilai elemennya (dalam pengurutan ini, indeks array akan disusun ulang) Bentuk umum: bool sort(array \$array [,int \$sort_flags = SORT_REGULAR])
asort()	Untuk mengurutkan (<i>ascending</i>) elemen-elemen pada array asosiatif Bentuk umum: bool asort(array \$array [,int \$sort_flags = SORT_REGULER])
ksort()	Fungsi khusus untuk mengurutkan (<i>ascending</i>) elemen array berdasarkan kunci (<i>key</i>) atau indeksnya Bentuk umum: bool ksort(array \$array [,int \$sort_flags = SORT_REGULER])
rshot()	Fungsi ini akan mengurutkan (<i>descending</i>) array berdasarkan nilai elemennya (dalam pengurutan ini, indeks array akan disusun ulang) Bentuk umum: bool rsort(array \$array [,int \$sort_flags = SORT_REGULER])
arsort()	Untuk mengurutkan (<i>descending</i>) elemen-elemen pada array asosiatif Bentuk umum: bool arsort(array \$array [,int \$sort_flags = SORT_REGULER])
krsort()	Fungsi khusus untuk mengurutkan (<i>descending</i>) elemen array berdasarkan kunci (<i>key</i>) atau indeksnya Bentuk umum: bool krsort(array \$array [,int \$sort_flags = SORT_REGULER])