Chapter 6

Array

6.1 Pengertian Array

Array sering dijumpai dalam kehidupan sehari-hari, terutama yang berkaitan dengan manipulasi table-tabel angka, atau dalam bentuk struktur data matematis yang disebut dengan vector dan matriks.

Array adalah suatu bentuk struktur data yang menampung satu atau lebih dari satu data yang sejenis (bertipe data sama), yang diwakili oleh satu nama variabel. Array sering juga disebut larik.

Setiap elemen atau anggota array dapat dikenali atau diakses melalui suatu indeks. Array berdimensi satu disebut vector, array berdimensi dua disebut dengan matriks, dan berdimensi lebih dari dua disebut tensor.

6.2 Membuat dan Mengakses Array

Untuk membuat/menciptakan suatu array digunakan fungsi `array()`

Contoh:

- (i) `$warna = array("merah","kuning","hijau","biru");`
- (ii) Atau dengan cara menyebut nomor elemen array sebagai berikut:
`$warna[0]="merah";`
`$warna[1]="kuning";`
`$warna[2]="hijau";`
`$warna[3]="biru";`
- (iii) Atau dengan cara tanpa menyebutkan nomor elemen array sebagai berikut:
`$warna[]="merah";`
`$warna[]="kuning";`
`$warna[]="hijau";`
`$warna[]="biru";`

Seperti yang dinyatakan di atas, bahwa setiap elemn pada array dibedakan berdasarkan nomor indeks. Jika nomor indeks tidak ditentukan, maka secara default data pertama akan diidentifikasi oleh PHP dengan indeks (nomor elemen array)=0, data kedua=1, ketiga=2, dan seterusnya. Array bernomor indeks disebut array berindeks, tetapi PHP juga mengenal array yang setiap elemennya dapat diacu dengan indeks berupa string yang disebut dengan array asosiatif.

Penamaan suatu array sama seperti penamaan variabel yang diawali dengan tanda dollar (\$) diikuti nama, kemudian kurung siku buka ([]) dan kurung siku tutup (]). Nama array digunakan untuk mengakses array, misalnya untuk mengakses elemen kedua dari array `$warna` maka dituliskan `$warna[1]`. Lengkapnya sebagai berikut:

	<code>\$warna[0]</code>	<code>\$warna[1]</code>	<code>\$warna[2]</code>	<code>\$warna[3]</code>
<code>\$warna</code>	Merah	Kuning	Hijau	Biru

6.3 Array Asosiatif

PHP memperkenankan indeks array berupa string. Ilustrasinya sebagai berikut:

	Jakarta	Jambi	Jayapura	Jember
\$kode	021	0741	089	033

Nama arraynya \$kode dengan indeks Jakarta, Jambi, Jayapura dan Jember. Dengan demikian, array \$kode dapat dituliskan sebagai berikut:

```
$kode["Jakarta"]="021";
```

```
$kode["Jambi"]="0741";
```

Kalau dengan menggunakan fungsi array(), ilustrasi di atas dapat ditulis sebagai berikut:

```
$kode=array("Jakarta"=>"021", "Jambi"=>"0741", "Jayapura"=>"089", "Jember"=>"033");
```

6.3.1 Mengambil Isi Array

Untuk mengambil isi array, anda bisa menggunakan notasi:

```
$nama_array[indeks]
```

```
Contoh print($warna[1]);
```

Akan menampilkan isi elemen kedua pada array warna.

6.3.2 Mengetahui Jumlah elemen Array

PHP menyediakan fungsi bernama count() yang berguna untuk mendapatkan jumlah elemen array. Fungsi ini memerlukan argument array bersangkutan. Sebagai contoh:

```
$musik=array("jazz", "blues", "fusion");
```

```
Printf("Jumlah elemen music adalah %d", count($musik));
```

6.4 Penggabungan Array

Beberapa array dapat digabung menjadi satu array yang lebih besar. Misalkan array A dengan 10 elemen akan digabungkan dengan array B dengan 15 elemen, tentu saja gabungannya berupa array C yang memiliki elemen 25. Untuk menggabungkan array, dalam PHP digunakan fungsi array_union() atau array_merge().

6.5 Fungsi Pengurutan pada Array

Beberapa aplikasi memerlukan data yang berurut, baik dari kecil ke besar (ascending) maupun dari besar ke kecil (descending). Banyak teknik yang digunakan untuk mengurutkan array seperti insertion sort, merge sort, dan bubble sort. Berikut ini adalah prinsip dari bubble sort:

1. Andaikan ada 5 elemen dalam array [10,8,3,5,4]
2. Mula-mula ambil indeks 1 sebagai patokan A[1]=10;
3. Bandingkan isi A[1] dengan A[2], A[3], A[4] dan A[5]
4. Bila A[1]>A[2] maka tukar tempat sehingga A[1]=8 ->[8,10,3,5,4]
5. Bila A[1]>A[3] maka tukar tempat sehingga [a1]=3 ->[3,10,8,5,4]
6. Bila A[1]>A[4] maka tukar tempat, tapi tidak terjadi
7. Bila A[1]>A[5] maka tukar tempat, tapi tidak terjadi
8. Sekarang ambil indeks 2 sebagai patokan A[2]
9. Bandingkan A[2] dengan A[3], A[4], dan A[5]
10. Hasilnya adalah [3,4,10,8,5]
11. Sekarang ambil indeks 3 sebagai patokan A[3]=10
12. Bandingkan A[3] dengan A[4],A[5]
13. Hasilnya adalah [3,4,5,10,8]
14. Sekarang ambil indkes 4 sebagai patokan A[4]=10
15. Bandingkan A[4] dengan A[5]
16. Hasilnya adalah [3,4,5,8,10]
17. Proses pengurutan selesai.

Proses pengurutan di atas, secara algoritma memerlukan dua buah indeks. Indeks pertama sebagai patokan yang bergerak dari 1 hingga N-1 (1 sampai 4 pada contoh di atas) dan indeks kedua sebagai pembanding yang selalu bergerak dari posisi indeks patokan +1 hingga posisi terakhir N(mulai dari 2 sampai 5). Bubble sort has worst-case and average complexity both $O(n^2)$, where n is the number of items being sorted

Fungsi pengurutan dalam array merupakan fungsi yang telah disediakan dalam PHP untuk mengurutkan elemen atau mengurutkan indeksnya. Sintaks dan fungsi-fungsi pengurutan adantara lain:

`arsort(array)` : mengurutkan elemen array secara menurun

`asort` : mengurutkan elemen array secara menaik

`asort` : mengurutkan indeks array secara menurun. Sangat berguna untuk pengurutan pada array asosiatif.

Contoh

```
<html>
<head>
<title>Pengurutan Array</title>
</head>
<body>
<?php
$kota[0]="Jakarta";
```

```
?>
</body>
</html>
```

6.6 Fungsi POP, PUSH dan SHIFT dalam array

Fungsi pop, push dan shift dapat diberlakukan pada array layaknya operasi pada stack(tumpukan). Sintaks dan fungsi tersebut adalah:

Array_pop(array) : Menghapus elemen terakhir dari satu array

Array_push(array, data1,...) : menambah satu atau beberapa elemen di akhir dari satu array.

Array_shift(array) : menghapus elemen pertam dari suatu array

Array_unshift(array, data1,...) : menambah satu atau beberapa elemen di awal suatu array.

PHP menyediakan fungsi bernama in_array() yang berguna untuk memeriksa apakah suatu nilai berada dalam array atau tidak. Bentuk pemanggilan fungsi ini:

In_array(yang_dicari, nama_array)

Hasil berupa TRUE kalau data yang dicari ada pada nama_array.

Contoh:

```
<html>
<head>
<title>Pemeriksaan keberadaan data</title>
</head>
<body>
<?php
$tumpukan[0]="A";
$tumpukan[1]="B";
$tumpukan[2]="C";
$tumpukan[3]="D";
Printf("Apakah D ada ditumpukan: ? %s <br>",in_array("D", $tumpukan)?"Ya" : "Tidak");
?>
</body>
</html>
```

6.7 Operasi dengan Pointer Internal

Setiap array memiliki pointer internal yang berguna untuk menunjuk elemen sekarang. PHP menyediakan sejumlah fungsi yang terkait dengan pointer seperti tabel berikut:

Fungsi	Keterangan
current	Menghasilkan nilai elemen sekarang yang ada dalam array. Sintaks: current(nama_array)
each	Menghasilkan indeks dan nilai elemen sekarang dan memajukan pointer ke elemen berikutnya Sintaks: each(nama_array)
end	Meletakkan pointer internal ke elemen terakhir pada array dan menghasilkan nilai elemennya Sintaks: end(nama_array)
key	Menghasilkan indeks milik elemen yang sedang ditunjuk oleh pointer internal Sintaks: key(nama_array)

pos	Menghasilkan nilai elemen sekarang Sintaks: pos(nama_array)
prev	Menghasilkan nilai elemen yang berada sebelum posisi pointer dan meletakkan pointer ke lokasi tersebut Sintaks: prev(nama_array)
reset	Melatakan pointer ke elemen pertama dan menghasilkan nilai elemen tersebut Sintaks: reset(nama_array)
next	Meletakkan pointer ke elemen berikutnya dan menghasilkan nilai elemen tersebut Sintaks: next(nama_array)

Skrip berikut dapat digunakan untuk memahami operasi dengan berbagai fungsi yang dibahas pada tabel di atas:

```
<html>
<head>
<title>Operasi pointer internal pada array</title>
</head>
<body>
<?php
$data["A"]="Sangat baik";
$data["B"]="Baik";
$data["C"]="Cukup";
$data["D"]="Gagal";
//contoh penggunaan current dan key
Print("<B>Setelah pendefinisian array</B><br>");
$indeks=key($data);
$nilai=current($data);
Print("Indeks sekarang adalah : $indeks <br>");
Print("Nilai elemen sekarang adalah : $nilai<br>");

//contoh penggunaan end
$nilai=end($data);
$indeks=key($data);
Print("<B>Setelah pemanggilan end</B><br>");
Print("Indeks sekarang adalah : $indeks <br>");
Print("Nilai elemen sekarang adalah : $nilai<br>");
?>
</body>
</html>
```

6.8 Array Multidimensi

Salah satu struktur data array yang juga banyak digunakan dalam berbagai aplikasi adalah matiks atau array 2D (dua dimensi), satu untuk menunjukkan baris dan yang lainnya menunjukkan kolom. Susunan angka berikut ini menunjukkan matriks 4x5 (4 baris dan 5 kolom).

```
10    4    4    3    14
```

23	5	6	1	15
45	7	7	3	12
32	8	5	5	13

Untuk mengisi suatu matriks berdimensi 4x5 dilakukan baris demi baris mulai dari baris 1 dengan mengisi kolom 1 sampai dengan kolom 5, kemudian pindah ke baris 2 dan kolom 1 sampai dengan kolom 5, dan seterusnya.

Array tiga dimensi sering disebut dengan array dari bentuk ruang. Jadi, array yang lebih dari satu dimensi sering disebut array multidimensi. Artinya, PHP juga mendukung array multidimensi.

Perhatikan contoh array multidimensi pada script berikut:

```
<html>
<head>
<title>Array Multidimensi</title>
</head>
<body>
<?php
$matrik=array(0=>array(1,2,3),1=>array(4,5,6),2=>array(7,8,9));
Print "Matrik <br>";
For ($i=0;$i<=2;$i++){
    For($j=0;$j<=2;$j++){
        Print($matrik[$i][$j])." ";
    }
    Print "<br>";
}
?>
</body>
</html>
```