

PHP Array

Oleh : Freddy Wicaksono, M.Kom

Array dalam PHP adalah struktur data yang digunakan untuk menyimpan banyak nilai dalam satu variabel. Array dapat menyimpan berbagai jenis data, termasuk string, integer, objek, atau bahkan array lain (array multidimensi). PHP mendukung dua tipe array utama: **array numerik** (diindeks dengan angka) dan **array asosiatif** (diindeks dengan string).

Jenis Array:

1. **Array Numerik**: Array dengan indeks numerik yang otomatis dimulai dari 0 atau dapat diubah sesuai kebutuhan.
2. **Array Asosiatif**: Array dengan indeks berupa string atau kunci tertentu yang ditentukan oleh pengguna.
3. **Array Multidimensi**: Array yang berisi array lainnya, memungkinkan penyimpanan data dalam bentuk tabel atau matriks.

Array Numerik

Array Numerik di PHP adalah jenis array yang menggunakan indeks berupa angka untuk mengakses elemen-elemen di dalamnya. Indeks pada array numerik dimulai dari 0 secara default dan dapat berlanjut sesuai urutan atau dapat juga diubah secara manual.

Ciri-ciri Array Numerik:

- **Indeks numerik**: Array ini menggunakan angka (integer) sebagai indeks untuk setiap elemen.
- **Indeks dimulai dari 0** secara default, tetapi bisa dimulai dari angka lain jika diinginkan.
- **Dapat berisi berbagai tipe data**: Seperti string, integer, atau bahkan array lain (array multidimensi).

10 Contoh Array Numerik di PHP:

1. Array Numerik Sederhana:

```
$warna = array("Merah", "Biru", "Hijau");  
echo $warna[0]; // Output: Merah
```

2. Array dengan Indeks Numerik yang Ditetapkan:

```
$angka = array(1 => "Satu", 2 => "Dua", 3 => "Tiga");  
echo $angka[2]; // Output: Dua
```

3. Array dengan Angka sebagai Elemen:

```
$angka = array(10, 20, 30, 40, 50);  
echo $angka[3]; // Output: 40
```

4. Array dengan Elemen String:

```
$buah = array("Apel", "Jeruk", "Mangga");  
echo $buah[1]; // Output: Jeruk
```

5. Array dengan Indeks Otomatis:

```
$hewan = array("Kucing", "Anjing", "Burung");  
echo $hewan[2]; // Output: Burung
```

6. Menambahkan Elemen ke Array Menggunakan `array_push()`:

```
$buah = array("Apel", "Pisang");  
array_push($buah, "Mangga", "Jeruk");  
print_r($buah); // Output: Array ( [0] => Apel [1] => Pisang [2] => Mangga  
[3] => Jeruk )
```

7. Mengubah Indeks dalam Array Numerik:

```
$angka = array(10, 20, 30);  
$angka[1] = 25; // Mengubah elemen kedua menjadi 25  
echo $angka[1]; // Output: 25
```

8. Menggunakan `foreach` untuk Iterasi Array:

```
$warna = array("Merah", "Biru", "Hijau");  
foreach ($warna as $value) {  
    echo $value . " "; // Output: Merah Biru Hijau  
}
```

9. Array Numerik dengan Elemen Campuran:

```
$data = array("Bola", "Sepakbola", "Tenis");  
$data[3] = "Badminton"; // Menambahkan elemen baru dengan indeks 3  
print_r($data); // Output: Array ( [0] => Bola [1] => Sepakbola [2] =>  
Tenis [3] => Badminton )
```

10. Array Numerik Multidimensi:

```
$kelas = array(  
    array("Budi", "Andi"),  
    array("Dewi", "Rani")  
);  
echo $kelas[1][0]; // Output: Dewi
```

Array Asosiatif

Array Asosiatif di PHP adalah jenis array yang menggunakan **kunci** (key) berupa **string** atau **angka** yang ditentukan secara eksplisit oleh pengguna, alih-alih menggunakan indeks numerik otomatis seperti pada array numerik. Kunci ini memungkinkan kita untuk mengakses elemen-elemen array dengan lebih mudah berdasarkan nama atau label yang lebih deskriptif.

Ciri-ciri Array Asosiatif:

- **Indeks berbentuk string atau kunci yang ditentukan sendiri**, bukan angka otomatis.
- Memungkinkan untuk memberikan label atau nama untuk elemen, yang membuat kode lebih mudah dipahami.
- Elemen dalam array bisa berupa berbagai tipe data (string, integer, array lainnya, dll.).

10 Contoh Array Asosiatif di PHP:

1. Array Asosiatif Sederhana:

```
$harga_buah = array("Apel" => 5000, "Jeruk" => 7000, "Mangga" => 6000);  
echo $harga_buah["Jeruk"]; // Output: 7000
```

2. Array Asosiatif dengan Kunci String dan Nilai Numerik:

```
$umur = array("John" => 25, "Doe" => 30, "Jane" => 22);  
echo $umur["Doe"]; // Output: 30
```

3. Array Asosiatif dengan Nilai Array:

```
$kontak = array(  
    "John" => array("email" => "john@example.com", "telepon" =>  
        "123456789"),  
    "Doe" => array("email" => "doe@example.com", "telepon" => "987654321")  
);  
echo $kontak["John"]["email"]; // Output: john@example.com
```

4. Menambahkan Elemen ke Array Asosiatif:

```
$produk = array("Laptop" => 15000000, "Handphone" => 5000000);  
$produk["Tablet"] = 3500000;  
print_r($produk); // Output: Array ( [Laptop] => 15000000 [Handphone] =>  
5000000 [Tablet] => 3500000 )
```

5. Mengubah Nilai Elemen dalam Array Asosiatif:

```
$harga_barang = array("Baju" => 100000, "Sepatu" => 150000);  
$harga_barang["Baju"] = 120000; // Mengubah harga Baju menjadi 120000  
echo $harga_barang["Baju"]; // Output: 120000
```

6. Array Asosiatif dengan Kunci Tipe Lain (Integer):

```
$produk = array(1 => "Susu", 2 => "Teh", 3 => "Kopi");  
echo $produk[2]; // Output: Teh
```

7. Menggunakan foreach untuk Iterasi Array Asosiatif:

```
$siswa = array("Nama" => "Budi", "Usia" => 20, "Kelas" => "A");  
foreach ($siswa as $key => $value) {  
    echo $key . ": " . $value . "\n"; // Output: Nama: Budi, Usia: 20,  
    Kelas: A  
}
```

8. Array Asosiatif dengan Data Campuran:

```
$karyawan = array(
    "ID" => 101,
    "Nama" => "Budi",
    "Posisi" => "Manager",
    "Gaji" => 7000000
);
echo $karyawan["Nama"]; // Output: Budi
```

9. Array Asosiatif dengan Menggunakan `array_merge()` :

```
$array1 = array("A" => "Merah", "B" => "Biru");
$array2 = array("C" => "Hijau", "D" => "Kuning");
$hasil = array_merge($array1, $array2);
print_r($hasil); // Output: Array ( [A] => Merah [B] => Biru [C] => Hijau
[D] => Kuning )
```

10. Array Asosiatif dengan Fungsi `array_keys()` dan `array_values()` :

```
$produk = array("Laptop" => 15000000, "Handphone" => 5000000);
$kunci = array_keys($produk); // Mengambil kunci array
$nilai = array_values($produk); // Mengambil nilai array
print_r($kunci); // Output: Array ( [0] => Laptop [1] => Handphone )
print_r($nilai); // Output: Array ( [0] => 15000000 [1] => 5000000 )
```

Array Multidimensi

Array Multidimensi di PHP adalah array yang memiliki lebih dari satu dimensi. Ini berarti bahwa elemen-elemen di dalam array tersebut dapat berupa array lagi, membentuk struktur array yang lebih kompleks, seperti tabel, matriks, atau struktur data lebih tinggi. Array multidimensi biasanya digunakan untuk menyimpan data yang terorganisir dalam bentuk baris dan kolom.

Ciri-ciri Array Multidimensi:

- **Array dalam Array:** Elemen-elemen dalam array multidimensi bisa berupa array lain, yang mengarah pada struktur yang lebih dalam (bisa lebih dari dua dimensi).
- **Penggunaan dalam Struktur Data:** Biasanya digunakan untuk menyimpan data yang membutuhkan dua atau lebih parameter untuk diakses, seperti data tabel atau matriks.
- **Dapat menggunakan indeks numerik atau asosiatif** untuk mengakses elemen di dalamnya.

10 Contoh Array Multidimensi di PHP:

1. Array Multidimensi dengan Indeks Numerik:

```
$matriks = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
echo $matriks[1][2]; // Output: 6
```

2. Array Multidimensi dengan Indeks Asosiatif:

```
$siswa = array(
    "John" => array("Usia" => 20, "Kelas" => "A"),
    "Doe" => array("Usia" => 22, "Kelas" => "B")
);
echo $siswa["Doe"]["Usia"]; // Output: 22
```

3. Matriks 3x3:

```
$matriks = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
echo $matriks[2][1]; // Output: 8
```

4. Array Multidimensi dengan Data Campuran:

```
$data = array(
    "orang1" => array("Nama" => "Budi", "Umur" => 25),
    "orang2" => array("Nama" => "Andi", "Umur" => 30)
);
echo $data["orang1"]["Nama"]; // Output: Budi
```

5. Array Multidimensi dengan Indeks Numerik dan Asosiatif:

```
$produk = array(
    "Laptop" => array("Harga" => 10000000, "Stok" => 50),
    "Handphone" => array("Harga" => 5000000, "Stok" => 100)
);
echo $produk["Handphone"]["Harga"]; // Output: 5000000
```

6. Array Multidimensi dengan Data Angka:

```
$nilai = array(
    array(75, 80, 85),
    array(70, 75, 78),
    array(90, 88, 85)
);
echo $nilai[2][0]; // Output: 90
```

7. Array Multidimensi dengan Penggunaan foreach :

```
$produk = array(
    array("ID" => 1, "Nama" => "Laptop", "Harga" => 10000000),
    array("ID" => 2, "Nama" => "Handphone", "Harga" => 5000000)
);
foreach ($produk as $item) {
    echo $item["Nama"] . ": " . $item["Harga"] . "\n";
}
// Output:
// Laptop: 10000000
// Handphone: 5000000
```

8. Matriks 2x2 dengan Data Asosiatif:

```
$matriks = array(
    "baris1" => array("A" => 1, "B" => 2),
    "baris2" => array("A" => 3, "B" => 4)
);
echo $matriks["baris1"]["A"]; // Output: 1
```

9. Array Multidimensi dengan Data Pribadi:

```
$karyawan = array(
    array("Nama" => "Budi", "Jabatan" => "Manager", "Gaji" => 5000000),
    array("Nama" => "Andi", "Jabatan" => "Staff", "Gaji" => 3000000)
);
echo $karyawan[1]["Jabatan"]; // Output: Staff
```

10. Array Multidimensi untuk Menyimpan Jadwal Pelajaran:

```
$jadwal = array(
    "Senin" => array("Matematika", "Bahasa Indonesia", "IPA"),
    "Selasa" => array("Bahasa Inggris", "Sejarah", "Pendidikan Agama"),
    "Rabu" => array("Seni", "Olahraga", "Matematika")
);
echo $jadwal["Senin"][2]; // Output: IPA
```

Kesimpulan:

- **Array Numerik** di PHP adalah array yang menggunakan indeks berupa angka untuk mengakses elemen.
- Indeks dimulai dari 0 secara default, namun dapat ditetapkan sesuai kebutuhan.
- Elemen dalam array numerik dapat berupa string, angka, atau tipe data lainnya.
- Fungsi `array_push()` dapat digunakan untuk menambahkan elemen ke dalam array.
- **Array Asosiatif** memungkinkan penggunaan kunci (string atau integer) untuk mengakses elemen array, menjadikannya lebih deskriptif dan mudah dibaca.
- Data di dalam array asosiatif dapat berupa berbagai tipe, seperti string, angka, array, atau objek.
- **Fungsi** seperti `array_keys()`, `array_values()`, dan `array_merge()` sangat berguna untuk manipulasi array asosiatif.
- **Array Multidimensi** digunakan untuk menyimpan data yang lebih kompleks, seperti tabel atau matriks, yang membutuhkan dua atau lebih dimensi untuk diakses.
- Elemen dalam array multidimensi bisa berupa array lain, yang membuatnya sangat fleksibel untuk menyimpan berbagai jenis data.
- **Array Asosiatif** dapat digabungkan dengan array multidimensi untuk membuat struktur data yang lebih deskriptif.
- **Iterasi dengan `foreach`** sangat berguna untuk mengakses setiap elemen dalam array multidimensi, memungkinkan manipulasi data yang lebih efisien.

Latihan

Berikut ini aplikasi PHP yang menggunakan array untuk menyimpan data produk dalam daftar belanja. Pengguna dapat menambahkan produk baru, menghapus produk, dan memperbarui harga produk.

```
<?php
// Array untuk menyimpan daftar produk dan harga
$produk = array(
    "Laptop" => 10000000,
    "Handphone" => 5000000,
    "TV" => 3000000
);

// Fungsi untuk menambah produk
function tambahProduk($nama, $harga) {
    global $produk;
    $produk[$nama] = $harga;
}

// Fungsi untuk menghapus produk
function hapusProduk($nama) {
    global $produk;
    if (isset($produk[$nama])) {
        unset($produk[$nama]);
    } else {
        echo "Produk tidak ditemukan.<br>";
    }
}

// Fungsi untuk memperbarui harga produk
function updateHarga($nama, $hargaBaru) {
    global $produk;
    if (isset($produk[$nama])) {
        $produk[$nama] = $hargaBaru;
    } else {
        echo "Produk tidak ditemukan.<br>";
    }
}

// Menampilkan daftar produk
function tampilkanProduk() {
    global $produk;
    echo "<h3>Daftar Produk:</h3>";
    foreach ($produk as $nama => $harga) {
        echo "Produk: $nama, Harga: Rp " . number_format($harga, 0, ',', '.')
        . "<br>";
    }
}

// Cek apakah ada input dari pengguna (simulasi form submission)
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['tambah'])) {
        // Menambah produk
        tambahProduk($_POST['produk'], $_POST['harga']);
    } elseif (isset($_POST['hapus'])) {
        // Menghapus produk
        hapusProduk($_POST['produk']);
    } elseif (isset($_POST['update'])) {
        // Memperbarui harga produk
        updateHarga($_POST['produk'], $_POST['harga']);
    }
}

// Menampilkan produk terbaru setelah operasi
```

```
tampilkanProduk();
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Daftar Belanja Produk</title>
</head>
<body>
  <h1>Daftar Belanja Produk</h1>

  <form method="POST">
    <h3>Tambah Produk Baru</h3>
    Nama Produk: <input type="text" name="produk" required><br>
    Harga Produk: <input type="number" name="harga" required><br>
    <button type="submit" name="tambah">Tambah Produk</button>
  </form>

  <form method="POST">
    <h3>Hapus Produk</h3>
    Nama Produk: <input type="text" name="produk" required><br>
    <button type="submit" name="hapus">Hapus Produk</button>
  </form>

  <form method="POST">
    <h3>Update Harga Produk</h3>
    Nama Produk: <input type="text" name="produk" required><br>
    Harga Baru: <input type="number" name="harga" required><br>
    <button type="submit" name="update">Update Harga</button>
  </form>

</body>
</html>
```

Praktikum:

Aplikasi Sistem Pengelolaan Inventaris Barang

Buatlah aplikasi PHP untuk mengelola inventaris barang. Aplikasi ini akan menyimpan data barang beserta kuantitasnya menggunakan array. Pengguna dapat menambahkan barang baru, memperbarui kuantitas barang, dan menghapus barang dari inventaris.

Fitur Aplikasi:

- 1. Menampilkan Daftar Inventaris Barang:
 - Buatlah array yang berisi nama barang dan kuantitasnya, misalnya:
 - Barang: "Pensil", Kuantitas: 100
 - Barang: "Buku", Kuantitas: 50
 - Barang: "Penghapus", Kuantitas: 75
 - Tampilkan daftar barang beserta kuantitasnya.
- 2. Menambahkan Barang Baru:
 - Buat form untuk menambahkan barang baru ke dalam inventaris.
 - Barang baru harus memiliki nama dan kuantitas yang dimasukkan oleh pengguna.

- Tambahkan barang baru ke dalam array setelah pengguna mengirimkan data.

3. **Memperbarui Kuantitas Barang:**

- Buat form untuk memperbarui kuantitas barang.
- Pengguna harus memasukkan nama barang dan kuantitas baru. Jika barang ada dalam inventaris, perbarui kuantitas barang tersebut.

4. **Menghapus Barang:**

- Buat form untuk menghapus barang berdasarkan nama barang yang dimasukkan oleh pengguna.
- Jika barang ada di dalam array, hapus barang tersebut.

5. **Menampilkan Daftar Inventaris Terbaru:**

- Setelah melakukan operasi, tampilkan daftar inventaris terbaru dengan kuantitas yang diperbarui.