

## Array - Part I

## array() | []

Mendefinisikan array

```
$arr = array(); // Semua versi PHP
$arr = []; // php >= 5.4
```

Mengakses array

```
$arr = array('foo', 'bar');
echo $arr[1]; // bar
$arr = array('a' => 'foo', 'b' => 'bar');
echo $arr['a']; // foo
```

## array\_push(arr, val1, ...) | array\_pop(arr)

array\_push: menambahkan elemen pada akhir array

```
$arr = array('foo', 'bar');
array_push($arr, 'alfa', 'beta');
```

array\_pop: menghapus elemen terakhir array

```
$arr = array('foo', 'bar');
array_pop($arr); // $arr: Array ([0]=>foo)
```

## array\_unshift (arr, val) | array\_shift (arr)

array\_unshift: menambahkan elemen pertama array

```
$arr = array('foo', 'bar');
array_unshift($arr, 'alfa');
// $arr: Array ([0]=>alfa [1]=>foo [2]=>bar)
```

array\_shift: menghapus elemen pertama array

```
$arr = array('foo', 'bar');
array_shift($arr); // $arr: Array ([0]=>bar)
```

## asort (arr) | arsort (arr)

asort: array sort, mengubah urutan value menjadi ascending

```
$arr = array('a' => 'foo', 'b' => 'bar');
asort($arr); // $arr: Array ([b]=>bar [a]=>foo)
```

arsort: array reverse sort, urutan value menjadi descending

```
$arr = array('bar', 'foo');
arsort($arr); // $arr: Array ([0]=>foo [1]=>bar)
```

## ksort (arr) | krsort (arr)

ksort = key sort, mengubah urutan key secara ascending

```
$arr = array('b' => 'bar', 'a' => 'foo');
ksort($arr); // $arr: Array ([a]=>foo [b]=>bar)
```

krsort = key reverse sort, urutan key menjadi descending

```
$arr = array('a' => 'foo', 'b' => 'bar');
krsort($arr); // $arr: Array ([b]=>bar [a]=>foo)
```

## array\_keys (arr) | array\_values (arr)

array\_keys: mengambil key dari array, array\_values: mengambil value dari array

```
$arr = array('a' => 'foo', 'b' => 'bar');
$key = array_keys ($arr); // Array ([0]=>a [1]=>b)
$key = array_values ($arr); // Array([0]=>foo [1]=>bar)
```

## max (arr) | min (arr)

max mengambil value maksimal, min value minimal

```
$arr = array ('a'=>1, 'b'=>2, 'c'=>3);
max($arr); // 3
min($arr); // 1
```

## count (arr)

Menghitung banyaknya elemen array

```
$arr = array('foo', 'bar');
echo count($arr); //2
```

## join (arr, separator = '')

Menggabung value array menjadi string

```
$arr = array('a' => 'foo', 'b' => 'bar');
$string = join ($arr, ', '); // foo, bar
```

## unset (arr[index])

Menghapus elemen array berdasar index

```
$arr = array('foo', 'bar');
unset($arr[1]); // $arr: Array ( [0]=>foo )
```

## array\_search (string, arr)

Mencari index berdasarkan value

```
$arr = array('a' => 'foo', 'b' => 'bar');
$key = array_search('bar', $arr); // $key: b
```

## array\_map (callback, arr)

Memanipulasi value dari array

```
$arr = array('foo', 'bar');
$upper = array_map('strtoupper', $arr);
// $upper: Array ([0]=>FOO [1]=>BAR)
```

Dengan fungsi sendiri:

```
$harga = array(1000, 2000);
$neto = array_map('neto', $harga);
function neto($arr_value) {
    return $arr_value * 0.5;
}
// $neto: Array ( [0] => 500 [1] => 1000 )
```

## Array - Part II

### list (arr)

Assign variabel pada value dari array

```
$arr = array ('merah', 'mobil');  
list ($warna, $kendaraan) = $arr;  
echo $warna; // merah
```

```
list (, $kendaraan) = $arr;  
echo $kendaraan; // mobil
```

### in\_array (str, arr) | key\_exists (str, arr)

in\_array: cek elemen array

```
$arr = array('bar', 'foo');  
if (in_array('foo', $arr)) { echo 'exists' }
```

key\_exists: cek key array

```
$arr = array('a' => 'foo', 'b' => 'bar');  
if (key_exists('b', $arr)) { echo 'exists' }
```

### array\_merge(arr, arr) | array(arr) + array(arr)

array\_merge: menggabung array, hasil indexed array

```
$arr = array('foo', 'bar');  
$arr2 = array('bar', 'alfa');  
$merge = array_merge($arr, $arr2);  
// Array ([0]=>foo [1]=>bar [2]=>bar [3]=>alfa)
```

Operator + untuk menggabung array berdasarkan index

```
$arr = array('bar', 'foo');  
$arr2 = array('a' => 'alfa');  
$arr3 = $arr + $arr2;  
// $arr3: Array ([0]=>bar [1]=>foo [a]=>alfa)  
  
$arr = array('bar', 'foo');  
$arr2 = array('alfa', 'beta', 'charlie');  
// $arr3: Array ([0]=>bar [1]=>foo [2]=>charlie)
```

### is\_array (arr)

Tes apakah argumen berbentuk array

```
$arg = 'string';  
if (is_array($arg)) {  
    echo 'array';  
} else {  
    echo 'bukan array';  
} // bukan array
```

## Jenis Array

Jenis Array: Indexed, associative, dan multidimensi.  
Array dapat digambarkan seperti tabel yang memiliki dua kolom:

### Jenis Array

key	value
0	Mobil
1	Sepeda
2	Truk
3	Motor
4	Bus

Indexed Array

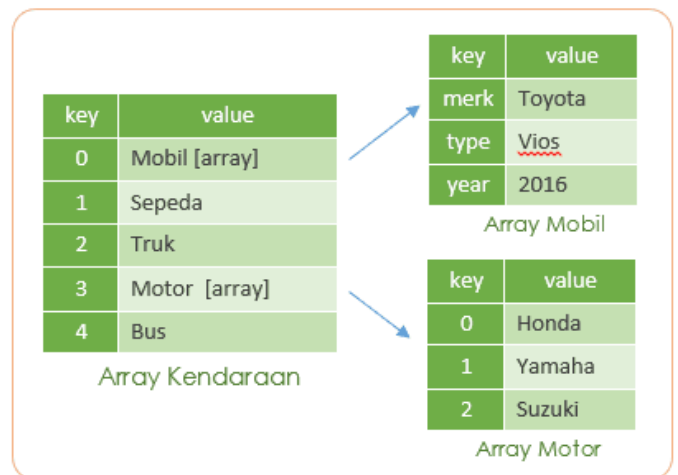
key	value
id	786
pass	123
role	admin
active	1
level	1

Associative Array

```
$kendaraan[1] // Sepeda  
$user['pass'] // 123
```

Value dari array juga bisa berbentuk array (menjadi key dari array yang lain), misal:

### Jenis Array



Multidimensional Array

Pada contoh diatas, value Mobil dan Motor menjadi key dari array mobil dan array motor, bentuk ini disebut Array Multidimensi (Multidimensional Array)

```
$kendaraan = array( array('merk'=>'Toyota',  
                           'type' => 'Vios',  
                           'year' => 2016  
                       ),  
                    'Sepeda',  
                    'Truk',  
                    array('Honda', 'Yamaha',  
                           'Suzuki'),  
                    'Bus',  
                    );  
$kendaraan[0]['merk']; // Toyota
```