

PHP

Array, Class, dan Objek

SMK Puragabaya
Rekayasa Perangkat Lunak



PENDAHULUAN

ARRAY

Array merupakan koleksi atau kumpulan data yang disimpan dalam satu variabel. Ini artinya tipe data array mampu menampung banyak data. Pada pemrograman PHP dan JavaScript contohnya, Array tidak mengharuskan valuenya memiliki tipe data yang sama.

A. Membuat array

Gunakan fungsi **array()** untuk membuat variabel dengan nilai array.

```
$dataArray = array(); // array kosong

// array dengan 3 nilai
$makanan = array("Batagor", "Pop Ice", "Pisgor");

// array dengan 2 nilai
$minuman = array("Pop Ice");
```

B. Mencetak data array

Gunakan fungsi **print_r()** untuk mencetak key (indeks) dan value array.

```
$hewan = array("Kucing", "Kelinci");
print_r($hewan);
```

Output :

Array ([0] => Kucing [1] => Kelinci)

Catatan : **[0]** dan **[1]** dinamakan key dari sebuah array. Ini artinya indeks ke-0 dari array tersebut memiliki value Kucing dan indeks ke-1 memiliki value Kelinci.

Gunakan tag HTML <pre> jika ingin hasilnya dicetak dalam format tabulasi.

```
$hewan = array("Kucing", "Kelinci");  
echo "<pre>";  
print_r($hewan);  
echo "</pre>";
```

Output :

Array

```
(  
    [0] => Kucing  
    [1] => Kelinci  
)
```

Catatan : Tag <pre> memungkinkan teks ditampilkan dalam bentuk **preformatted** sehingga dapat menampilkan karakter spasi, baris baru (enter), atau tabulasi (tab) yang berulang.

C. Mengakses data array

Data array dapat diakses dengan memanggil indeksinya. Indeks array dimulai dari nol (0).

```
$hewan = array("Kucing", "Kelinci");  
echo $hewan[0];
```

Output :

Kucing

D. Mengubah nilai elemen array

Layaknya variabel dengan tipe data primitif (string, number), nilai elemen array dapat diubah dengan memberikan nilai baru menggunakan operator assignment. Perbedaannya, indeks elemen array yang akan diubah harus disebutkan.

```
$hewan = array("Kucing", "Kelinci");  
$hewan[0] = "Pisang";  
echo $hewan[0]; // nilai indeks ke-0 sekarang menjadi Pisang
```

Output :

Pisang

E. Mengakses data array dengan perulangan FOR

Data array dapat diakses seluruhnya menggunakan kontrol perulangan. Perulangan **for** memungkinkan untuk menyebutkan indeks array secara berurutan.

```
$hewan = array("Kucing", "Kelinci", "Kura-Kura");  
  
$total_elemen_array = count($hewan);  
for ($i = 0; $i < $total_elemen_array; $i++) {  
    echo $hewan[$i] . "<br>";  
}
```

Output :

Kucing

Kelinci

Kura-Kura

F. Mengakses data array dengan perulangan WHILE

Data array dapat diakses seluruhnya menggunakan kontrol perulangan **while**. Perbedaannya dengan perulangan **for** adalah penambahan **iterator** dilakukan di akhir perulangan.

```
$hewan = array("Kucing", "Kelinci", "Kura-Kura");  
  
$total_elemen_array = count($hewan);  
$i = 0; // dimulai dari elemen array ke-0
```

```
while ($i < $total_elemen_array) {  
    echo $hewan[$i] . "<br>";  
    $i++;  
}
```

Output :

Kucing
Kelinci
Kura-Kura

G. Mengakses data array dengan perulangan FOREACH

Data array dapat diakses seluruhnya menggunakan kontrol perulangan **foreach**. Perulangan **foreach** memungkinkan untuk mengakses value dari array tanpa menyebutkan indeksinya.

```
$hewan = array("Kucing", "Kelinci", "Kura-Kura");  
  
// dibaca: untuk setiap nilai elemen array $hewan sebagai $value  
foreach ($hewan as $value) {  
    echo $value . "<br>";  
}
```

Output :

Kucing
Kelinci
Kura-Kura

H. Menambahkan data ke dalam array

Syntax :

```
array_push(array, value1, value2, ...);
```

Contoh :

```
$merk_mobil = array("Avanza", "Tesla", "Inova");  
array_push($merk_mobil, "Mustibisha");
```

Catatan : Cetak elemen array untuk melihat hasilnya.

I. Menghapus elemen array dengan fungsi ARRAY_SPLICE()

Syntax :

```
array_splice(array, start, delete_length);
```

Contoh : Menghapus data array merk_mobil pada indeks ke-1 sebanyak 1 elemen.

```
$merk_mobil = array("Avanza", "Tesla", "Inova");  
array_splice($merk_mobil, 1, 1);
```

Hasilnya, tersisa 2 elemen array :

```
"Avanza", "Inova"
```

Catatan : Cetak elemen array untuk melihat hasilnya.

J. Menyisipkan elemen array

Data array dapat ditambahkan pada indeks tertentu menggunakan **array_splice()** dengan menambahkan parameter terakhir.

Syntax :

```
array_splice(array, start, delete_length, value1, value2, ...);
```

Contoh 1 : Menyisipkan nilai **Mustibisha** pada indeks ke 1.

```
$merk_mobil = array("Avanza", "Tesla", "Inova");  
array_splice($merk_mobil, 1, 0, "Mustibisha");
```

Hasilnya, tersisa 4 elemen array :

```
"Avanza", "Mustibisha", "Tesla", "Inova"
```

Catatan : Cetak elemen array untuk melihat hasilnya.

Contoh 2 : Menghapus 1 buah elemen array pada indeks ke-1 kemudian menyisipkan nilai **Mustibisha** dan **Tayo**.

```
$merk_mobil = array("Avanza", "Tesla", "Inova");  
array_splice($merk_mobil, 1, 1, "Mustibisha", "Tayo");
```

Hasilnya, tersisa 4 elemen array :

```
"Avanza", "Mustibisha", "Tayo", "Inova"
```

Catatan : Cetak elemen array untuk melihat hasilnya.

K. Menghapus elemen array terakhir

Syntax :

```
array_pop(array);
```

Contoh : Menghapus data array merk_mobil pada indeks ke-1 sebanyak 1 elemen.

```
$merk_mobil = array("Avanza", "Tesla", "Inova");  
array_pop($merk_mobil);
```

Hasilnya, tersisa 2 elemen array :

```
"Avanza", "Tesla"
```

Catatan : Cetak elemen array untuk melihat hasilnya.

CLASS & OBJEK

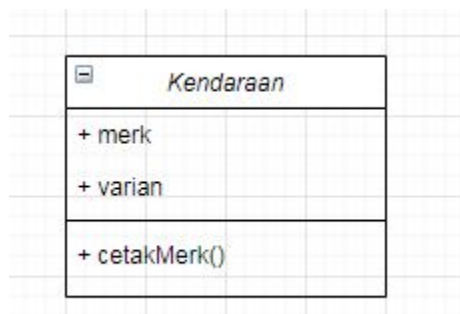
Dalam pemrograman berorientasi objek (PBO), class adalah tipe data baru berupa kerangka program untuk menginstansiasi (membuat) objek. Class dapat dibayangkan sebagai *blueprint* sebuah objek. Atribut apa saja yang dimiliki dan method apa saja yang dapat dikerjakan sebuah objek nantinya akan didefinisikan di dalam class ini.

A. Mendefinisikan class

Syntax :

```
class Nama_Kelas {  
    ... isi atribut / method sebuah kelas ...  
}
```

Contoh :



Class Diagram **Kendaraan**

Catatan : Class Diagram adalah salah satu model UML (Unified Modeling Language) yang digunakan untuk mendefinisikan struktur sebuah class. Class Diagram menampilkan nama class, atribut, method, dan visibilitas atribut/method.

Visibilitas	Simbol	Cakupan Akses (Scope)
Public	+	Dapat diakses oleh seluruh bagian program
Protected	#	Hanya dapat diakses oleh dirinya sendiri dan turunannya
Private	-	Hanya dapat diakses oleh dirinya sendiri

```
class Kendaraan {
    public $merk = "Avanza";
    public $varian = "So Black";
    public function cetak_merk() {
        // kode untuk mencetak merk kendaraan
    }
}
```

Perlu diperhatikan bahwa penamaan class menggunakan aturan **PascalCase**, nama class harus diawali dengan huruf kapital.

B. Membuat objek dari class

Dalam pemrograman berorientasi objek, objek adalah hasil instansiasi dari sebuah class. Untuk membuat objek baru dari sebuah class, gunakan keyword **new**.

Syntax :

```
new nama_objek;
```

Contoh : Membuat objek baru dari class Kendaraan.

```
class Kendaraan {
    public $merk = "Avanza";
    public $varian = "So Black";
    public function cetak_merk() {
        // kode untuk mencetak merk kendaraan
    }
}

$mobil = new Kendaraan;
```

C. Mengakses properti sebuah objek

Properti dari sebuah objek (atribut dan method) dapat diakses dengan menuliskan tanda panah setelah nama objek diikuti dengan nama properti objek.

Syntax :

```
nama_objek->properti_objek;
```

Contoh : Mengakses atribut **merk** dan method **cetak_merk()** pada objek **\$mobil**.

```
class Kendaraan {  
    public $merk = "Avanza";  
    public $varian = "So Black";  
  
    public function cetak_merk() {  
        echo "Merk mobil ini rahasia";  
    }  
}  
  
$mobil = new Kendaraan;  
  
echo $mobil->merk;           // hasilnya Avanza  
$mobil->cetak_merk();        // hasilnya Merk mobil ini rahasia
```

D. Memberi nilai awal pada atribut objek

Pada contoh sebelumnya, nilai atribut objek dituliskan langsung pada class seperti kode berikut :

```
class Kendaraan {  
    public $merk = "Avanza";  
    public $varian = "So Black";  
}
```

Pada kasus input data, kita perlu memberi nilai sesuai data masukan dari user. Fungsi **constructor** dapat digunakan untuk mencapai hal ini. Constructor adalah fungsi pada class yang akan dieksekusi saat pembuatan objek baru yang dipanggil menggunakan keyword **new**.

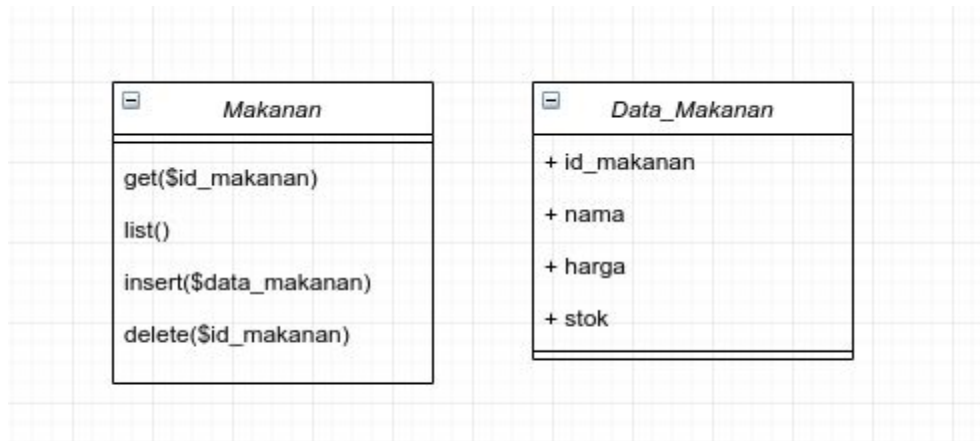
Syntax :

```
function __construct(argumen1, argumen2, ...) {  
    // kode yang akan dieksekusi ketika objek dibuat  
}
```

Contoh : Memberikan nilai "Avanza" pada atribut merk dan nilai "Cool Pink" pada atribut varian objek mobil.

```
class Data_Kendaraan {  
    public $merk;  
    public $varian;  
  
    public function __construct($m, $v) {  
        $this->merk = $m;  
        $this->varian = $v;  
    }  
}  
  
$mobil = new Kendaraan("Avanza", "Cool Pink");  
  
echo $mobil->merk; // hasilnya Avanza  
echo "<br>";  
echo "Variannya adalah : " . $mobil->varian; // hasilnya Cool Pink
```

Latihan



Class Diagram **Makanan** dan **Data_Makanan**

Implementasikan **class diagram** di atas ke dalam pemrograman PHP. Ada pun spesifikasi method dari class **Makanan** adalah sebagai berikut :

- **get(\$id_makanan)**

Program menerima parameter **id_makanan**, kemudian mencetak data makanan sesuai id_makanan yang dicari.

- **list()**

Program menampilkan daftar data makanan.

- **insert(\$data_makanan)**

Program menerima input data objek makanan, kemudian menambahkan data makanan.

- **delete(\$id_makanan)**

Program menerima input **id_makanan**, kemudian menghapus data makanan sesuai id_makanan yang dicari.

== Bagian 1 ==

Inisialisasi dan pemberian nilai awal.

Petunjuk : Gunakan method **insert()** pada class **Makanan** untuk menyisipkan data ke array **gudang**.

1. Buat sebuah array untuk menampung data makanan dengan nama **gudang**.
2. Tambah data di bawah ini ke array **gudang** :

id_makanan	nama	harga	stok
0	Batagor	Rp. 5.000,00	20
1	Pop Ice	Rp. 3.000,00	20
2	Mie Goreng	Rp. 7.000,00	50

== Bagian 2 ==

Menampilkan dan memanipulasi data **array of object**.

Petunjuk : Gunakan tag **<table>..</table>** untuk membuat tampilan tabel.

3. Tampilkan seluruh data makanan dalam **gudang** dengan tampilan tabel.
4. Ada orang yang memborong Pop Ice untuk kawinan, hapus data makanan dengan nama **Pop Ice**.
5. Tampilkan kembali seluruh data makanan. Pastikan data Pop Ice sudah terhapus.
6. Tambahkan data makanan di bawah ini sebelum **Mie Goreng** :

id_makanan	nama	harga	stok
1	Piatos Keju Bakar Garing	Rp. 10.000,00	1

7. Tampilkan kembali seluruh data makanan. Pastikan data yang baru telah ditambahkan sebelum Mie Goreng



== Bagian 3 ==

Seleksi data

Petunjuk : Gunakan kontrol program **IF ... ELSE** untuk menyeleksi data makanan.

8. Tampilkan data makanan dengan id_makanan = "2"
9. Tampilkan data makanan dengan nama = "Batagor" atau stok = "20"

■ ■ ■