

Konsep Dasar OOP

Apa perbedaan class dan object

Oleh : Putera Kahfi

1. Pendahuluan

Bagi yang baru belajar tentang OOP tentu akan mengalami kebingungan tentang perbedaan antara class dan object. Pada bahasan ini akan di jelaskan apa perbedaan antara class dan object, dan bagaimana implementasi nya di OOP.

2. Class

Analogi yang cukup tepat untuk menggambarkan class adalah sebuah sketsa/blueprint mobil, sebelum mobil di produksi ada tahapan membuat blueprint , blueprint ini menggambarkan tentang bagaimana bentuk mobil tersebut, di sketsa tersebut dijelaskan tentang warna, jumlah pintu, jenis mesin, jumlah roda, inilah yang sering di sebut dengan **property**. Dimana property adalah sebuah pengenal yang menjadi ciri object tersebut.

Selain itu di jelaskan tentang hal-hal yang bisa di lakukan oleh mobil tersebut seperti mengerem, berjalan menggunakan roda, membunyikan klakson, membuka pintu, maju, mundur, berhenti, semua hal tersebut kita sebut sebagai method. Dimana method sendiri adalah menjelaskan tentang bagaimana sebuah object berperilaku (behaviour).

Jadi class adalah **konsep yang di gunakan untuk menjelaskan bagaimana suatu object tersebut akan di produksi/di buat.**

Atau kita bisa menyebutnya sebagai sebuah **blueprint dari object itu**

sendiri.

3. Object

Setelah sebelumnya kita tahu bahwa class adalah blueprint atau konsep yang menjelaskan tentang bagaimana sebuah object di bentuk nantinya. Maka langkah selanjutnya adalah memahami object. Telah di jelaskan sebelumnya bahwa class merupakan blueprint sebuah object, maka object itu sendiri adalah **bentuk concrete/nyata** dari blueprint class tersebut.

Jadi setelah blueprint mobil selesai, artinya kita sudah tahu jenis mobil yang akan di produksi jenis apa? Mesinnya menggunakan apa, warnanya apa saja. Langkah selanjutnya adalah memproduksi mobil tersebut.

Maka object adalah, **bentuk concrete dari sebuah blueprint** , dan bisa di buat lebih dari satu. blueprint mobilnya mungkin cuman satu, tapi mobil yang di produksi bisa lebih dari satu, dengan berbagai macam warna yang berbeda.

4. Implementasi

Langkah selanjutnya adalah bagaimana mengimplementasikan sebuah class, object, pada contoh ini yang di gunakan adalah bahasa pemrograman php.

Seperti telah di jelaskan sebelumnya bahwa class terdiri dari property, method, dimana properti menggambarkan bagaimana sebuah object di kenali (bisa dari warna, jumlah pintu,no rangka mesin, ukuran ban dll). Dan method adalah apa-apa yang bisa object lakukan (mengerem, berjalan, klakson, buka pintu dll)

Maka contoh class Mobil dan implementasinya adalah seperti berikut :

```
<?php

/**
 * Description of Mobil
 *
 * @author dwc
 */
class Mobil
{

    /** property */
    private $warna;
    private $rangkaMesin;
    private $statEngine = 'mati';
    private $tahunProduksi;
    private $label;

    /** attribute */

    public function setLabel($label){
        $this->label = $label;
    }

    public function setWarna($warna)
    {
        $this->warna = $warna;
    }

    public function setNoRangka($noRangka){
        $this->rangkaMesin =$noRangka;
    }

    public function setTahunProduksi($tahunProduksi)
    {
        $this->tahunProduksi = $tahunProduksi;
    }

    public function nyalakan()
    {
        echo "brum brum brum (ceritanya mesinnya nyala) <br/>";
        $this->statEngine = 'nyala';
    }

    public function matikan()
    {
        echo "mobil sudah mati <br/>";
    }
}
```

```

        $this->statEngine = 'mati';
    }

    public function maju()
    {
        if ($this->statEngine == 'nyala') {
            echo "mobil maju";
        } else {
            echo "nyalain mesinnya dulu om <br/>";
        }
    }

    public function getInfo()
    {
        $infoMobil = "INFO MOBIL ".$this->label;
        $infoMobil .= "<br/>=====";
        $infoMobil .= "<br/> WARNA :".$this->warna;
        $infoMobil .= "<br/> TAHUN: ".$this->tahunProduksi;
        $infoMobil .= "<br/> NO RANGKA MESIN: ". $this->rangkaMesin;
        $infoMobil .= "<br/>=====<br/>";
        echo $infoMobil;
    }
}

```

// Buat object dari class mobil

```

$mobilSaya = new Mobil();
$mobilSaya->setWarna('Merah');
$mobilSaya->setTahunProduksi('2015');
$mobilSaya->setNoRangka('45145-35-345');
$mobilSaya->setLabel('Mobil saya nih');
$mobilSaya->getInfo();

$mobilSaya->maju();

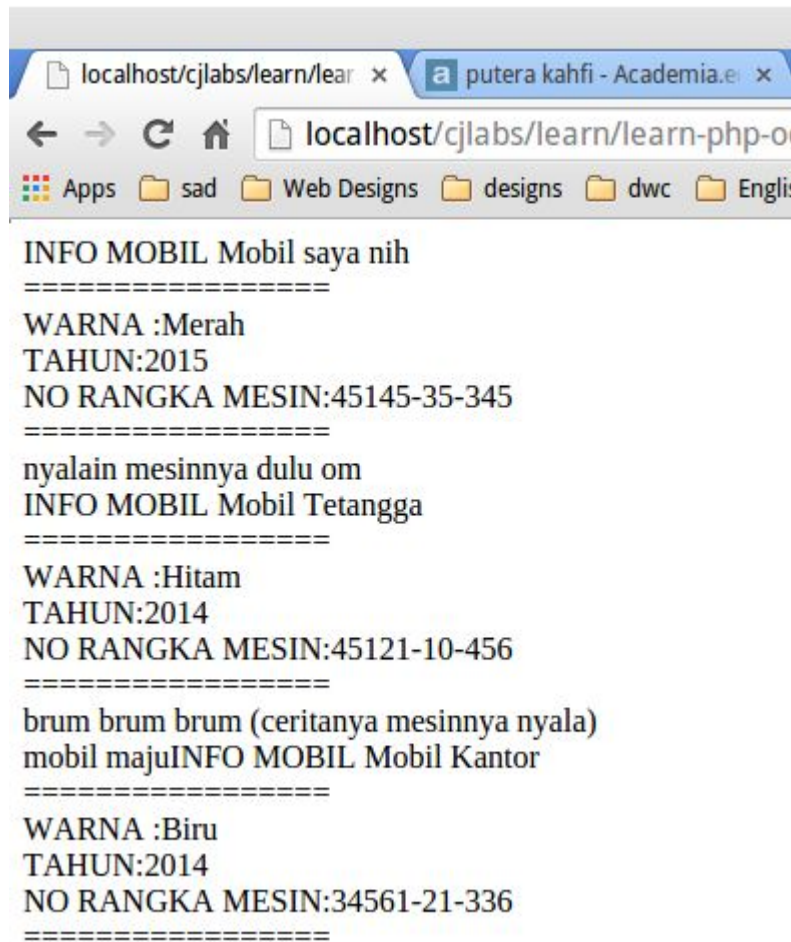
$mobilTetangga = new Mobil();
$mobilTetangga->setLabel('Mobil Tetangga');
$mobilTetangga->setWarna('Hitam');
$mobilTetangga->setTahunProduksi('2014');
$mobilTetangga->setNoRangka('45121-10-456');
$mobilTetangga->getInfo();
$mobilTetangga->nyalakan();
$mobilTetangga->maju();

$mobilKantor = new Mobil();
$mobilKantor->setLabel('Mobil Kantor');
$mobilKantor->setWarna('Biru');
$mobilKantor->setTahunProduksi('2014');
$mobilKantor->setNoRangka('34561-21-336');
$mobilKantor->getInfo();

```

?>

Jika kode program tersebut di jalankan di web browser maka akan ditampilkan info tentang detail tentang masing-masing mobil (dari contoh diatas berarti ada 3 info mobil).



Gambar 1. Hasil implementasi class mobil

Seperti yang anda lihat, bahwa dengan satu class/blueprint mobil, dapat di hasilkan beberapa objek mobil dengan spesifikasi yang berbeda-beda. Itulah kenapa objek di sebut sebagai **bentuk concrete dari sebuah blueprint** , dan bisa di buat lebih dari satu.

5. Kesimpulan

Setelah membaca paparan sekilas tentang perbedaan Class dan object, maka anda tentu dapat membedakan perbedaan kedua istilah OOP tersebut. Dimana class merupakan sebuah blueprint/konsep tentang bagaimana sebuah object tersebut nanti jadinya. Dan object sendiri adalah bentuk concrete/instance dari class dan bisa lebih dari satu dan dengan property yang bisa berbeda-beda.