**#11 OOP DASAR pada PHP Static Keyword**

Kita lihat dulu kasus ini ya :

class Mobil {

    public $nama, $merk, $warna, $kecepatanMaksimal, $jumlahPenumpang;

    public function tambahKecepatan() {

    }

    public function kurangiKecepatan() {

    }

    public function gantiTransmisi() {

    }

}

$avanza = new Mobil();

Jadi sebuah class yg di instasiasi menjadi object, object nya itu property $avanza itu object nya. Terus kita punya kasus yg lain saya punya class Mahasiswa bserta property dan method didalam nya :

class Mahasiswa {

    private $nama, $umur, $ipk;

    public function \_\_construct($nama, $umur, $ipk) {

**$this**->nama = $nama;

**$this**->umur = $umur;

**$this**->ipk = $ipk;

    }

    public function getNama() {

        return **$this**->nama;

    }

}

$mhs1 = new Mahasiswa('Keyla'. 19, 2.5);

echo $mhs1->getNama();

itu ga aneh ya karena sama-sama ada class dan objet jadi disini kita tau class merupakan template dari object ketika kita membuat sebuah object atau isitilah nya mengisntasiasi. Nah disitu lah kita bisa menggunakan property dan method yg ada didalam class.

Nah tapi sebetul nya kita bisa mengakses property dan method dalam konteks class itu makud nya kita bisa mengakses property dan method tanpa melakukan instansiasi dari class tersebut. Jadi ga harus di instasiasi juga bisa gitu jadi kita langsung akses property dan method nya.

Jadi kita bisa lakukan menggunakan Static Keyword, kita bisa bikin Static Property dan Static Keyword, untuk supaya kita bisa akses pada konteks class.

**CONTOH NYA :**

Adad

Kita buat class ContohStatic yg didalam nya property static dan method static, nah ada keyword nya itu static gini :

class ContohStatic {

    public static $angka = 1;

    public static function halo() {

        return "Hallo";

    }

}

Kalo kita mau akses $angka 1 itu biasa nya harus instasiasi dulu ke object ya. Nah karena dia public dan ada juga keyword static nya kita bisa akses secara langsung seperti variable dan function biasa.

Cukup panggil aja class nya terus kalo mua property nya tamabhin titik 2 nya gini :

class ContohStatic {

    public static $angka = 1;

    public static function halo() {

        return "Hallo";

    }

}

echo ContohStatic::$angka;

echo "<br>";

echo ContohStatic::halo();

jadi kita pake nya titik 2 titik 2, bukan lagi tanda panah, itu caranya untuk method static dan property static.

Gimana kalo di method halo() itu saya mau ambil property $angka, kalo biasanya kan ( $this->angka ) nah kalo pake $this itu hanya bisa jika class nya di instasiasi ke sebuah object tapi karena kita pake static gabisa pake $this.

Kita ganti dengan keyword nya itu self:: begini hasil nya

class ContohStatic {

    public static $angka = 1;

    public static function halo() {

        return "Hallo " . self::$angka . " kali.";

    }

}

Nah jika kita tambahkan ++ di angka nya gini :

class ContohStatic {

    public static $angka = 1;

    public static function halo() {

        return "Hallo " . self::$angka++ . " kali.";

    }

}

echo ContohStatic::$angka;

echo "<br>";

echo ContohStatic::halo();

echo "<br>";

echo ContohStatic::halo();

echo "<br>";

echo ContohStatic::halo();

nah hasil nya nanti yg panggil nya pertama halo 1 kali, keduanya “halo 2 kali” dan seterus nya gitu.

**UNTUK APA STATIC KEYWORD >>**

Static Keyword

* Member yg terkait dengan class, bukan dengan object. ( member itu property dan method, kalo mau akses dengan konteks class pake lah static keyword)
* Nilai static akan selalu tetap meskipun object di-instansiasi berulang kali. Nah ini yg menarik kita coba lihat ya :

class Contoh {

    public $angka = 1;

    public function halo() {

        return "Hallo " . **$this**->angka++ . " kali";

    }

}

$obj = new Contoh();

echo $obj->halo();

echo $obj->halo();

echo $obj->halo();

kita buat contoh nya sama seperti yg static keyword tapi ini yg versi oop biasa nya ya. Nah hasil nya sama kaya static keyword gaada yg aneh kan ya, angka nya juga ada ++ nya.

Nah sekarang saya buat object baru :

class Contoh {

    public $angka = 1;

    public function halo() {

        return "Hallo " . **$this**->angka++ . " kali" . "<br>";

    }

}

$obj = new Contoh();

echo $obj->halo();

echo $obj->halo();

echo $obj->halo();

echo "<hr>";

$obj2 = new Contoh();

echo $obj2->halo();

echo $obj2->halo();

echo $obj2->halo();

nah hasil nya sama saja $obj dan $obj2 ya, nah sekarang gimana kalo kita ganti $angka nya jadi static tapi function halo() biarkan saja biasa gini :

class Contoh {

    public static $angka = 1;

    public function halo() {

        return "Hallo " . self::$angka++ . " kali" . "<br>";

    }

}

$obj = new Contoh();

echo $obj->halo();

echo $obj->halo();

echo $obj->halo();

echo "<hr>";

$obj2 = new Contoh();

echo $obj2->halo();

echo $obj2->halo();

echo $obj2->halo();

Lihat kita hanya mengganti angka nya menjadi static tapi hasil nya itu berbeda dengan oop biasa hasil antara $obj dan $obj2 itu berbeda yg di sebelum nya itu hasil nya sama sekarang mereka berbeda hasil nya.

Jadi ternyata nilai nya tidak di reset jadi satu meskipun kita instasiasi object baru begitu ya kalo static.

Static Keyword

* Member yg terkait dengan class, bukan dengan object. ( member itu property dan method, kalo mau akses dengan konteks class pake lah static keyword)
* Nilai static akan selalu tetap meskipun object di-instansiasi berulang kali.
* Membuat kode kita seolah-olah menjadi ‘procedural’. (karena kita bisa langsung memanggil property dan method lewat class nya tanpa instasiasi).
* Biasanya digunakan untuk membuat fungsi bantuan atau helper ( karena gausah melakukan instansiasi langsung aja panggil property nya.
* Dan static ini juga biasa digunakan di class-class utility pada Framework