

PRAKTIKUM PEMROGRAMAN WEB LANJUT

PHP : CLASSES AND OBJECTS

1. Mengakses Property dan Method Parent Class

/* code1.php */

```
<?php

class smartphone {
    public function lihat_spec() {
        return "Spec smartphone: Samsung,
        Processor Snapdragon, Ram 1GB";
    }
}

class smartphone_dualsim extends smartphone {
    public function lihat_spec() {
        return "Spec smartphone dual sim: Asus,
        Processor Snapdragon, Ram 2GB";
    }
    public function lihat_spec_smartphone() {
        return parent::lihat_spec();
    }
}

$gadget_baru = new smartphone_dualsim();
echo $gadget_baru->lihat_spec();
echo "<br />";
echo $gadget_baru->lihat_spec_smartphone();
?>
```

/* code2.php */

```
<?php

class smartphone {
    public $merk = "Samsung";
    public function lihat_spec() {
        return "Spec smartphone: Samsung,
        Processor Snapdragon, Ram 1GB";
    }
}
```

```

class smartphone_dualsim extends smartphone {
    public $merk = "asus";
    public function lihat_spec() {
        return "Spec smartphone dual sim: Asus,
        Processor Snapdragon, Ram 2GB";
    }
    public function lihat_spec_smartphone() {
        return parent::lihat_spec();
    }
    public function lihat_merk_smartphone() {
        return parent::$merk;
    }
}

$gadget_baru = new smartphone_dualsim();
echo $gadget_baru->lihat_spec();
echo "<br />";
echo $gadget_baru->lihat_spec_smartphone();
echo "<br />";
echo $gadget_baru->lihat_merk_smartphone();
?>

```

Perbaiki code2 sehingga apabila dijalankan hasilnya sebagai berikut:

```

Spec smartphone dual sim: Asus, Processor Snapdragon, Ram 2GB
Spec smartphone: Samsung, Processor Snapdragon, Ram 1GB
Samsung

```

2. Mengakses Constructor dan Destructor Parent Class

/* code3.php */

```

<?php

class smartphone {
    public function __construct() {
        echo "Constructor dari class smartphone <br />";
    }
    public function __destruct() {
        echo "Destructor dari class smartphone <br />";
    }
}

```

```
class smartphone_dualsim extends smartphone {  
}  
  
class smartphone_singlesim extends smartphone_dualsim {  
}  
  
$gadget_baru = new smartphone_singlesim();  
echo "Belajar OOP PHP <br />";  
?>
```

/* code4.php */

```
<?php  
  
class smartphone {  
    public function __construct() {  
        echo "Constructor dari class smartphone <br />";  
    }  
    public function __destruct() {  
        echo "Destructor dari class smartphone <br />";  
    }  
}  
  
class smartphone_dualsim extends smartphone {  
    public function __construct() {  
        echo "Constructor dari class smartphone_dualsim <br />";  
    }  
    public function __destruct() {  
        echo "Destructor dari class smartphone_dualsim <br />";  
    }  
}  
  
class smartphone_singlesim extends smartphone_dualsim {  
    public function __construct() {  
        echo "Constructor dari class smartphone_singlesim <br />";  
    }  
    public function __destruct() {  
        echo "Destructor dari class smartphone_singlesim <br />";  
    }  
}  
  
$gadget_baru = new smartphone_singlesim();  
echo "Belajar OOP PHP <br />";  
?>
```

Perbaiki code4 sehingga apabila dijalankan hasilnya sebagai berikut:

Constructor dari class smartphone
Constructor dari class smartphone_dualsim
Constructor dari class smartphone_singlesim
Belajar OOP PHP
Destructor dari class smartphone_singlesim
Destructor dari class smartphone_dualsim
Destructor dari class smartphone

3. Static Property dan Static Method

/* code5.php */

```
<?php

class smartphone {
    public $merk;
    public $pemilik;

    // static property
    public static $harga_beli;

    //static method
    public static function beli_smartphone() {
        return "Beli smartphone";
    }
}

// set static property
smartphone::$harga_beli=3000000;
// get static property
echo "Harga beli : Rp".smartphone::$harga_beli;
echo "<br />";
// panggil static method
echo smartphone::beli_smartphone();
?>
```

/* code6.php */

```
<?php

class smartphone {
    public $merk;
    public $pemilik;

    // static property
    public static $harga_beli;

    //static method
    public static function beli_smartphone() {
        return "Beli smartphone seharga Rp".self::$harga_beli;
    }
}

// set static property
smartphone::$harga_beli=3000000;
// panggil static method
echo smartphone::beli_smartphone();
?>
```

/* code7.php */

```
<?php

class smartphone {
    protected static function beli_smartphone(){
        return "Beli smartphone baru";
    }
}

class smartphone_dualsim extends smartphone{
    private static function beli_smartphone_dualsim(){
        return "Beli smartphone dual sim baru";
    }

    public static function beli_semua(){
        //lengkapi baris kode berikut ini
    }
}

smartphone_dualsim::beli_semua();
?>
```

Lengkapi method beli semua() sehingga apabila dijalankan hasilnya sebagai berikut:

Beli smartphone baru
Beli smartphone dualsim baru

4. Konstanta Class dalam Pemrograman Objek

/* code8.php */

```
<?php  
  
class smartphone {  
    // buat konstanta  
    const RUPIAH = '3000000';  
}  
  
// panggil konstanta class  
echo "Harga smartphone saat ini = Rp. ".smartphone::RUPIAH;  
?>
```

/* code9.php */

```
<?php  
  
class smartphone {  
    // buat konstanta  
    const RUPIAH = '3000000';  
}  
  
$smartphone_obj = new smartphone();  
// panggil konstanta class  
echo "Harga smartphone saat ini = Rp. ".$smartphone_obj::RUPIAH;  
?>
```

/* code10.php */

```
<?php
class smartphone {
    // buat konstanta
    const DOLLAR = '12000';

    // buat method
    public function beli_smartphone($harga) {
        return "Beli smartphone baru, Rp. ".$harga*self::DOLLAR;
    }
}

$smartphone_obj = new smartphone();
echo $smartphone_obj->beli_smartphone(200);
?>
```

/* code11.php */

```
<?php
class smartphone {
    const DOLLAR = '13000';
}

class tablet extends smartphone {
    const DOLLAR = '12000';

    public function beli_smartphone($harga){
        //lengkapi baris kode berikut ini
    }

    public function beli_tablet($harga){
        return "Beli tablet baru, Rp ".$harga*self::DOLLAR;
    }
}

$smartphone_obj = new smartphone();

echo $smartphone_obj->beli_smartphone(200);
echo "<br />";
echo $smartphone_obj->beli_tablet(400);
?>
```

Perbaiki code11 dan lengkapi method beli_smartphone(\$harga) sehingga apabila dijalankan hasilnya sebagai berikut:

Beli smartphone baru, Rp .2600000

Beli Tablet baru, Rp .4800000

5. Final Method dan Final Class Pemrograman Objek

/* code12.php */

```
<?php
class smartphone {
    final public function lihat_spec() {
        return "Spec smartphone: Samsung,
        Processor Snapdragon, Ram 1GB";
    }
}

class smartphone_dualsim extends smartphone {
    public function lihat_spec() {
        return "Spec smartphone dual sim: Asus,
        Processor Snapdragon, Ram 2GB";
    }
}

$gadget_baru = new smartphone_dualsim();
?>
```

/* code13.php */

```
<?php
final class smartphone {
    function lihat_spec() {
        return "Spec smartphone: Samsung,
        Processor Snapdragon, Ram 1GB";
    }
}

class smartphone_dualsim extends smartphone {
}

$gadget_baru = new smartphone_dualsim();
?>
```


6. Abstract Class dan Abstract Method PHP

/* code14.php */

```
<?php

abstract class smartphone{
    // buat abstract method
    abstract public function lihat_spec();
}

class tablet extends smartphone{

    // implementasi abstract method
    public function lihat_spec(){
        return "Lihat Spec Tablet...";
    }

    // method 'biasa'
    public function beli_tablet(){
        return "Beli Tablet...";
    }
}

$tablet_obj = new tablet();
echo $tablet_obj->lihat_spec();
echo "<br />";
echo $tablet_obj->beli_tablet();
?>
```

/* code15.php */

```
<?php

// buat abstract class
abstract class smartphone{
    public static function hidupkan_komputer(){
        echo "Hidupkan Komputer";
    }
    // buat abstract method
    abstract public function lihat_spec();
    abstract public function lihat_processor();
    abstract public function lihat_harddisk();
    abstract public function lihat_pemilik();
}
```

```
class tablet extends smartphone{
    // .. isi class tablet
}

?>
```

Lengkapi code15 dan isi class tablesehingga apabila dijalankan hasilnya sebagai berikut:

```
Hidupkan Komputer
Lihat spec tablet...
Lihat processor tablet...
Lihat hardisk tablet...
Pemilik tablet adalah...
```

7. Object Interface Dalam Pemrograman Berbasis Objek

/ code16.php */*

```
<?php

interface mouse{
    const JENIS = "Laser Mouse";
    public function klik_kanan();
    public function klik_kiri();
}

interface mouse_gaming extends mouse{
    public function ubah_dpi();
}

interface keyboard{
    public function tekan_enter();
}

class laptop implements mouse{
    public function klik_kanan(){
        return "Klik Kanan...";
    }
    public function klik_kiri(){
        return "Klik Kiri...";
    }
    public function ubah_dpi(){
        return "Ubah settingan DPI mouse";
    }
}
```

```

class pc implements mouse, keyboard{
    public function klik_kanan(){
        return "Klik Kanan...";
    }
    public function klik_kiri(){
        return "Klik Kiri...";
    }
    public function ubah_dpi(){
        return "Ubah settingan DPI mouse";
    }
    public function tekan_enter(){
        return "Tekan Tombol Enter...";
    }
}

echo mouse::JENIS;
echo "<br />";

$laptop_obj = new laptop();
echo $laptop_obj->klik_kanan();
echo "<br />";
echo $laptop_obj->ubah_dpi();
echo "<br />";

$pc_obj = new pc();
echo $pc_obj->klik_kiri();
echo "<br />";
echo $pc_obj->tekan_enter();
?>

```

Buatlah perbedaan dan persamaan antara Interface Class dan Abstrak Class!

8. Polimorfisme dalam Pemrograman Objek PHP

/* code17.php */

```

<?php

// buat abstract class
abstract class komputer{
    // buat abstract method
    abstract public function booting_os();
}

```

```

interface mouse{
    public function double_klik();
}

class laptop extends komputer implements mouse{
    public function booting_os(){
        return "Proses Booting Sistem Operasi Laptop";
    }
    public function double_klik(){
        return "Double Klik Mouse Laptop";
    }
}

class pc extends komputer implements mouse{
    public function booting_os(){
        return "Proses Booting Sistem Operasi PC";
    }
    public function double_klik(){
        return "Double Klik Mouse PC";
    }
}

class chromebook extends komputer implements mouse{
    public function booting_os(){
        return "Proses Booting Sistem Operasi Chromebook";
    }
    public function double_klik(){
        return "Double Klik Mouse Chromebook";
    }
}

// buat objek dari class diatas
$laptop_baru = new laptop();
$pc_baru = new pc();
$chromebook_baru = new chromebook();

// buat fungsi untuk memproses objek
function booting_os_komputer($objek_komputer){
    return $objek_komputer->booting_os();
}

function double_klik_komputer($objek_komputer){
    return $objek_komputer->double_klik();
}

```

```
// jalankan fungsi
echo booting_os_komputer($laptop_baru);
echo "<br />";
echo double_klik_komputer($laptop_baru);
echo "<br />";
echo "<br />";

echo booting_os_komputer($pc_baru);
echo "<br />";
echo double_klik_komputer($pc_baru);
echo "<br />";
echo "<br />";

echo booting_os_komputer($chromebook_baru);
echo "<br />";
echo double_klik_komputer($chromebook_baru);
?>
```

Kegiatan!

1. Cobalah masing-masing kode.
2. Kerjakan masing-masing perintah di atas.
3. Submit ke github, dengan nama folder 11_Kelas_NIM.